

Trabajo Final de Grado

Grado en Ingeniería en Tecnologías Industriales

**GUÍA DE INSTALACIÓN, CONFIGURACIÓN Y
VALIDACIÓN DE SOFTWARE LIBRE PARA LA
SIMULACIÓN NUMÉRICA DE FLUIDOS (CFD)**

MEMORIA

Autor: Andrés Carrasco Sánchez
Director: Francesc Xavier Escaler Puigoriol
Convocatoria: Mayo 2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resumen

La simulación numérica es un conjunto de herramientas en continuo y rápido desarrollo durante estos últimos años. Tiene como objetivo reproducir comportamientos reales, a partir de parámetros otorgados por el usuario, que describen un problema a ser resuelto. En el caso de la fluidodinámica, se usa para estudiar la interacción de un fluido en una región y condiciones predefinidas.

En la actualidad existen una gran cantidad de *softwares* comerciales capaces de realizar esta función. Alternativamente, existen diversos softwares libres y de código abierto, basados en el sistema operativo Linux, con el objetivo de llevar a cabo la simulación numérica de fluidos (CFD). Concretamente, los softwares SALOME y OpenFOAM son unos de los más populares para este uso tanto en el ámbito académico como el industrial.

Este proyecto define la función de una simulación numérica y el procedimiento necesario para poder realizarla. Se ejecutarán dos simulaciones distintas mediante el uso de programas libres y de código abierto. Además, se documentará los pasos realizados en todo el proceso. El fin es crear una guía básica de uso de estos programas para casos sencillos. La guía comprende desde la instalación de dichos programas hasta los pasos que hay que seguir en el proceso de simulación para lograr los resultados obtenidos. Se contempla desde la creación de la geometría, su mallado, preparación, solución, hasta la obtención de resultados numéricos y gráficos. Estos resultados también se verificarán y se compararán con los obtenidos por un programa comercial (ANSYS). Una vez realizados los estudios, se valorará las ventajas e inconvenientes de su uso en torno a su facilidad de empleo o a las limitaciones que pueda presentar.

Sumario

SUMARIO	4
1. GLOSARIO	7
2. PREFACIO	11
2.1. Origen del proyecto y motivación	11
2.2. Conocimientos previos	11
3. INTRODUCCIÓN	13
3.1. Objetivos del proyecto	13
3.2. Alcance del proyecto	14
4. SIMULACIÓN NUMÉRICA	15
4.1. Proceso de simulación computacional de fluidos	17
4.1.1. Geometría.....	17
4.1.2. Mallado	18
4.1.2.1. Mallado en la proximidad a la pared: <i>inflation layer</i>	19
4.1.3. <i>Set-up</i>	22
4.1.4. <i>Solver</i>	23
4.1.5. Post-procesado.....	24
5. SOFTWARE UTILIZADO	26
6. GUÍA DE INSTALACIÓN DE LOS PROGRAMAS NECESARIOS PARA EL ESTUDIO	27
6.1.1. Instalación del sistema operativo.....	27
6.1.2. Instalación de SALOME.....	28
6.1.3. Instalación de OpenFOAM y Paraview.....	28
7. GUÍA DE USO PARA ESTUDIOS DE FLUJO INTERNO Y EXTERNO	29
7.1. Estudio comparativo de flujo interno	30
7.1.1. Geometría y mallado]	30
7.1.2. Set-up y solución	33
7.1.3. Post-procesado, resultados y conclusiones	37
7.2. Estudio comparativo de flujo externo	44
7.2.1. Geometría.....	44
7.2.2. Mallado	48
7.2.3. Set-up y solución	53
7.2.4. Post-procesado, resultado y conclusiones	57

8. VALORACIÓN DEL SOFTWARE LIBRE PARA EL USO EN SIMULACIÓN NUMÉRICA	64
9. IMPACTO AMBIENTAL	66
10. PLANIFICACIÓN	67
11. PRESUPUESTO	68
CONCLUSIONES	69
AGRADECIMIENTOS	71
BIBLIOGRAFÍA	73
Referencias bibliográficas	73
Bibliografía complementaria	74

1. Glosario

Análisis dimensional: Procedimiento por el cual se obtienen variables adimensionales a partir de otras proporcionadas en el problema. El objetivo de esta práctica es disminuir el número y complejidad de variables que intervienen en un fenómeno físico.

BIOS: *Basic Input/Output System*. Sistema que incorpora la configuración inicial del ordenador. Es el primer programa que se ejecuta a la hora de encender el ordenador, que tiene como objetivo iniciar, configurar, comprobar el buen funcionamiento del sistema y acceder al sistema operativo que contenga el dispositivo (si hay más de uno, el que esté primero en el orden de su lista).

CAD: *Computer Aided Design*, Diseño Asistido por Ordenador.

C_D : *Drag Coefficient*, coeficiente de arrastre. Parámetro adimensional que valora la resistencia que genera un cuerpo de unas características geométricas concretas cuando está en contacto con un fluido en movimiento. Este coeficiente depende del coeficiente de fricción (c_f , *skin friction coefficient*), causado por la viscosidad del fluido que se mueve en la región cercana a la superficie de contacto con el objeto; la forma del propio cuerpo, y el componente generado por la redirección del aire que impacta en el mismo. Se define por la Ecuación 1.

$$C_D = \frac{2 \cdot F_D}{\rho V^2 A}$$

Ecuación 1: Coeficiente de arrastre (C_D)

Donde:

- F_D : fuerza de arrastre
- ρ : densidad del fluido
- V : velocidad
- A : área de referencia

CFD: *Computational Fluid Dynamics*, Mecánica de Fluidos Computacional. Es la resolución mediante el uso de métodos numéricos de problemas de estudio del movimiento de fluidos.

C_L : *Lift Coefficient*, coeficiente de sustentación. Parámetro adimensional que valora la sustentación provocada por el contacto de un fluido con un cuerpo de características concretas. Se define por la Ecuación 2 (donde F_L es fuerza de sustentación).

$$C_L = \frac{2 \cdot F_L}{\rho V^2 A}$$

Ecuación 2: Coeficiente de sustentación (CL)

Número de Mach (M): Número adimensional que relaciona la velocidad del fluido con la velocidad de propagación del sonido en el mismo. En el caso que este número sea inferior a 0,3, el flujo se puede considerar incompresible (su densidad es constante). Se describe mediante la Ecuación 3.

$$M = \frac{V}{V_{\text{propagación sonido}}}$$

Ecuación 3: Número de Mach (M)

Número de Reynolds (Re): Es un número adimensional que caracteriza el comportamiento de un fluido. Se describe mediante la Ecuación 4:

$$Re = \frac{\rho V L}{\mu}$$

Ecuación 4: Número de Reynolds

Donde:

- ρ : densidad del fluido [kg/m^3]
- V : velocidad del fluido [m/s]
- L : longitud característica de la geometría de estudio [m]
- μ : viscosidad dinámica [$\text{Pa}\cdot\text{s}$]

Open Source software: Código abierto. Aquél programa que puede ser modificado en su totalidad, de forma que pueda disponer de un desarrollo continuo y sea actualizado y mejorado de forma constante.

Punto de estancamiento: Punto donde la velocidad del fluido es nula. Corresponde a un valor de presión elevado comparado con otros puntos de su alrededor.

RANS: *Reynolds-averaged Navier-Stokes equations*. Grupo de ecuaciones basadas en el desglose de las variables en sus valores medios y fluctuantes. Sirven para modelizar la turbulencia de un fluido.

Simulación de régimen estacionario: Estudio donde el campo de variables de cualquier punto del cuerpo es independiente del tiempo.

Software libre [1]: Aquel programa que respeta la libertad del que lo usa. No significa necesariamente que sea gratuito (aunque a menudo tiene relación), sino que el usuario dispone de la libertad de ejecución del programa con cualquier finalidad: poder estudiarlo y modificarlo, ser capaz de distribuirlo a terceros (tanto el programa original como modificado) y aprovechar las ventajas de la modificación para continuar desarrollándolo y mejorándolo. Se diferencia del *software* de código abierto por su finalidad. Mientras que en *software* de código abierto el objetivo último es la mejora continua del mismo, en el *software* libre se pretende que el usuario posea de total libertad. Aun así, la mayoría de programas *open source* son a la vez *software* libre. En el caso inverso, un *software* libre implica necesariamente que sea de código abierto.

Software comercial: Aquél programa que no cumple con las libertades mencionadas en el *software* libre.

Streamline: Línea de corriente. Línea que es tangente al vector velocidad en todos los puntos que la forman en un instante determinado.

2. Prefacio

2.1. Origen del proyecto y motivación

Actualmente la simulación numérica computacional es cada vez más usada en la industria con el fin de poder entender el comportamiento del producto deseado de estudio. Se suele realizar antes de poner en marcha la fabricación del prototipo, con el objetivo de poder prever y corregir posibles errores.

A nivel académico, es importante que los alumnos puedan disponer de todas las libertades para su aprendizaje. El conocimiento de este tipo de herramientas es útil tanto para el desarrollo profesional del alumno como en el campo de la investigación.

En estos sectores el poder simular este tipo de comportamientos es fundamental, por ello la existencia de programas de *software* libre es una necesidad.

Por estas causas, la motivación principal es dotar de una guía básica a cualquier usuario para el uso en casos simples de este tipo de programas.

2.2. Conocimientos previos

En este trabajo han sido aplicado conceptos vistos durante el Grado de Ingeniería en Tecnologías Industriales (GETI). Especialmente, gran parte de este trabajo es basado en conceptos explicados en la asignatura de Mecánica de Fluidos. Además, el uso del *software* comercial ANSYS, que es utilizado para la realización del estudio comparativo, también se ha enseñado de manera introductoria en esta asignatura.

Durante la elaboración del proyecto ha sido necesaria para la simulación conocimientos sobre el funcionamiento del sistema operativo Linux, en su distribución Xubuntu; SALOME, OpenFOAM y Paraview.

3. Introducción

3.1. Objetivos del proyecto

Actualmente en el departamento de Mecánica de Fluidos de la ETSEIB se dispone de un conjunto de *softwares* de simulación numérica de fluidos para uso académico. Entre ellos se usa ANSYS Fluent o CFX, *softwares* englobados en el entorno ANSYS, para el aprendizaje del alumnado. Este programa comercial engloba todo tipo de herramientas de simulación, no únicamente de fluidos.

Este *software* es uno de los más populares y usados para simulaciones. La interfaz de usuario destaca por su relativa sencillez de uso. Para el aprendizaje hay un gran número de tutoriales, muchos proporcionados por la propia compañía a través de guías o cursos online o presenciales en distintas ciudades del mundo.

No obstante, este programa es comercial y de un precio elevado para un gran sector de usuarios potenciales. Existen versiones para estudiantes, con un coste reducido, pero con ello también se reduce la capacidad computacional del programa. En muchas ocasiones esto conlleva a la limitación de su uso a casos sencillos o muy simplificados. Esto aleja la solución del caso real, en algunos casos de manera significativa.

Como alternativas existen *softwares* libres de simulación numérica de fluidos, y entre ellos OpenFOAM, que podría ser un recurso útil de aprendizaje para el alumnado de la asignatura. Su problema reside en el complicado aprendizaje, por una menor cantidad de guías y tutoriales paso a paso.

Se pretende crear una guía que registre el procedimiento necesario para la realización de una simulación mediante *software* libre. Se pretende abarcar la instalación de estos programas y el funcionamiento de cada uno de ellos para la realización de dos casos, uno de flujo interno y otro de externo, sometidos a condiciones diferentes. El objetivo es ser capaz de llegar a resultados de mapas de velocidad, presión líneas de corrientes y vectores como se es capaz en un software comercial (como ANSYS). Una vez logrado, se busca realizar una valoración sobre las ventajas o inconvenientes que pueden presentar estos tipos de programas respecto *softwares* comerciales.

Esta documentación de todo el proceso se realiza con el objetivo de facilitar el uso a personas que buscan realizar futuros trabajos con él. También se busca una introducción del *software* en las clases y prácticas de Mecánica de Fluidos.

3.2. Alcance del proyecto

Este trabajo contemplará la realización de casos sencillos en herramientas de simulación de OpenFOAM y ANSYS. El estudio persigue poder obtener unos resultados significativamente parecidos en un *software* libre y comercial. No contemplará un estudio profundo de la optimización del resultado para que sea lo más completamente fiel a un caso real posible, aunque se perseguirán unos resultados coherentes.

El fluido en todos los estudios será siempre un fluido newtoniano. Será el único fluido de la región estudiada, evitando la dificultad añadida del estudio de la interacción entre más de uno.

Los estudios que se harán serán de casos sencillos de flujos internos y externos, en los que la transferencia de calor no formará parte del problema (todo se considerará a una temperatura ambiente de $T=25^{\circ}\text{C}$) y la densidad del fluido será constante. Tampoco se tendrá en cuenta el posible efecto de la gravedad. La simulación será de régimen estacionario.

Todos los casos serán con un comportamiento de fluido bidimensional. Aun así, debido a que OpenFOAM no abarca estudios en bidimensionales, se hará un entorno tridimensional y se aplicarán condiciones descritas posteriormente para simplificarlo.

4. Simulación numérica

Existen tres métodos distintos para la obtención de datos de un estudio de mecánica de fluidos:

Por una parte, existen ciertos casos que se pueden resolver de manera directa de forma matemática, a partir de fórmulas. Estas fórmulas se basan en las ecuaciones en diferencias parciales de conservación de continuidad, cantidad de movimiento y energía, que describen el comportamiento del fluido. Para casos muy simplificados se puede encontrar solución a estas ecuaciones integrándolas, pero suponen una serie de limitaciones que no son prácticas para la mayoría de estudios realizados.

Las ecuaciones básicas son la Ecuación 5, Ecuación 6 y Ecuación 7) [2]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0$$

Ecuación 5: Ecuación diferencial de continuidad

$$\rho \frac{d\mathbf{V}}{dt} = -\nabla P + \rho \mathbf{g} + \nabla \cdot \tau_{ij}$$

Ecuación 6: Ecuación diferencial de cantidad de movimiento

$$\rho \frac{du}{dt} + P(\nabla \cdot \mathbf{V}) = \nabla \cdot (k\nabla T) + \Phi$$

Ecuación 7: Ecuación diferencial de la energía

Donde:

- ρ : densidad [kg/m³]
- t : tiempo [s]
- \mathbf{V} : velocidad [m/s]
- P : presión [Pa]
- g : gravedad [m/s²]
- τ : tensión [Pa]
- u : energía interna [J/kg]
- k : conductividad térmica [W/mK]

- T: temperatura [K]
- Φ : función de disipación viscosa (transferencia de energía del fluido a calor por fricción) [W/m^3]

Además necesitan de otras auxiliares, dependiendo de las condiciones del caso a resolver, que hagan que sea el sistema resultante tenga el mismo número de ecuaciones que de incógnitas. Por ejemplo, si se determina el fluido como newtoniano su tensión viene descrita por la Ecuación 8:

$$\tau_{xy} = -\mu \frac{dV_x}{dy}$$

Ecuación 8: Caracterización de la tensión para fluidos Newtonianos

Donde:

- μ : viscosidad dinámica del fluido [$\text{Pa}\cdot\text{s}$]
- dV_x/dy : gradiente de velocidad perpendicular a la tensión buscada [s^{-1}]

Estas ecuaciones sirven únicamente para flujos laminares. Para turbulentos (usando el enfoque de las ecuaciones RANS) existe un componente del campo de velocidades que fluctúa, debido al comportamiento caótico del fluido. Esto provoca unas tensiones adicionales llamadas tensiones turbulentas. Para relacionar estas tensiones con las variables de presión y velocidad se necesitan modelos de turbulencia. Son modelos semiempíricos, y la elección depende de las circunstancias, hecho que dificulta aún más la obtención de un resultado directo.

Por otra parte, el estudio se puede hacer de forma experimental, utilizando un modelo que simula a un prototipo. La relación entre las condiciones del modelo y prototipo se debe tener en cuenta con el fin de poder extrapolar los resultados del modelo al caso real. El problema de este método es la complicación para llevarlo a cabo. Es difícil experimentar dependiendo de las condiciones, y aun pudiéndolo hacer, el coste suele ser elevado. Además, los resultados experimentales que pueden ser medidos son limitados, relativamente poco flexibles a la hora de cambiar parámetros, y conllevan una acumulación de errores durante la realización del experimento, que pueden alejar el valor obtenido del real.

La tercera vía para poder obtener datos es la simulación computacional. La simulación computacional de fluidos (CFD) es una simulación numérica, es decir, es una implementación de un modelo matemático a un sistema físico de un estudio fluidodinámico. A diferencia del primer método, se caracteriza porque estos modelos se linealizan con el objetivo de conseguir soluciones en más tipos de casos de estudio diferentes. Esta forma

de obtención de resultados es mucho más económica y flexible que la experimentación, a la vez que puede resolver una mayor variedad de problemas que el método de resolución matemática directa.

Ningún método es exclusivo del otro, es decir, según las condiciones planteadas en un problema, el usuario puede escoger uno de ellos o varios, y así poder correlacionar los resultados y verificarlos.

Este tercer método necesita de una herramienta de simulación computacional, es decir, un programa o serie de programas que permitan la resolución del problema siguiendo este método.

4.1. Proceso de simulación computacional de fluidos

Para la obtención de datos mediante la simulación computacional de fluidos es necesario realizar el siguiente proceso:

4.1.1. Geometría

Primeramente se debe indicar cuál es el volumen de trabajo. Los softwares de simulación fluidodinámica establecen un espacio fijo, el cual es recorrido por el fluido (es decir, se usa una descripción Euleriana). Este volumen es especificado en forma y dimensiones a través del programa. Dependiendo de las herramientas de simulación es posible diseñar la geometría a través de coordenadas escritas directamente en el programa, o creadas (o importadas) a partir de un programa de CAD.

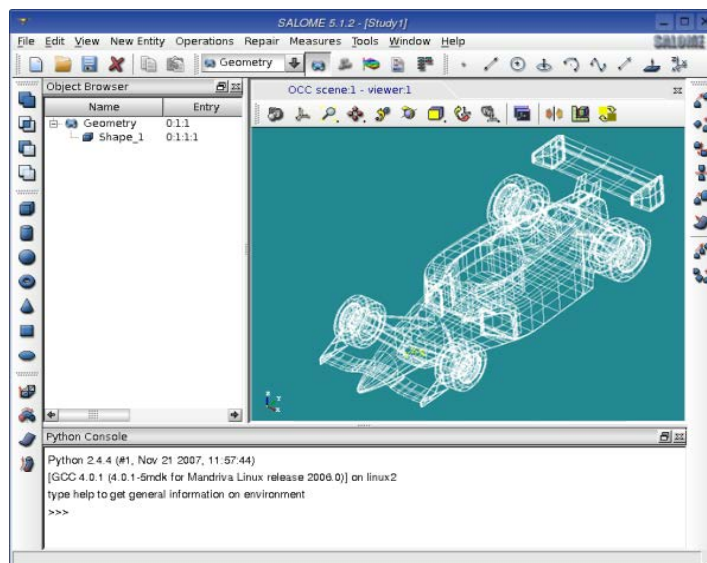


Figura 1: Geometría generada en SALOME [4]

4.1.2. Mallado

El proceso de mallado es el de división de la geometría descrita previamente para poder aplicar el Método de Volúmenes Finitos. Este método permite discretizar las ecuaciones a solucionar en los pequeños volúmenes o celdas. Para ello es importante elegir su estructura. Una malla no estructurada hará que la aplicación de las ecuaciones sea más complicada, por lo que la solución será más costosa de obtener (en términos de tiempo y de capacidad computacional necesaria). Con la malla estructurada sucede lo contrario, pero es más complicada de aplicar a geometrías complejas.

En algunos casos unos volúmenes demasiado irregulares, o una transición brusca de un volumen a otro pueden conllevar a dificultades a la hora de solucionar el problema, por lo que es importante prestar atención a los indicadores de calidad del mallado que ofrecen los distintos programas.

La decisión del tipo de elemento, y su tamaño influyen en la semejanza de la respuesta final a la real. Normalmente son parámetros que se pueden controlar en la herramienta de simulación.

El objetivo es buscar un mallado que consiga una relación óptima entre el realismo de los resultados y la cantidad de tiempo invertido en la simulación. Cuantos más elementos, más tiempo es necesario para la resolución de las ecuaciones, pero en más espacios son aplicadas las ecuaciones, por lo que el resultado final tiene una mayor resolución.

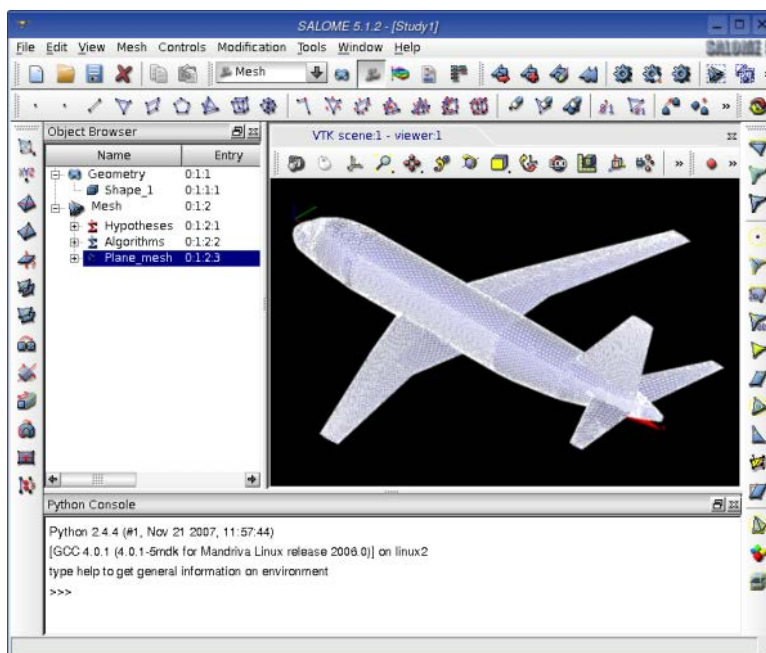


Figura 2: Geometría mallada en SALOME [4]

Posteriormente es común hacer estudios de sensibilidad de mallado. Estos consisten en ver la variación de variables de la solución dependiendo del número de elementos del mallado. Un mallado óptimo es uno a partir del cual los resultados no varían de forma significativa por aumentar el número de elementos y que son de un tiempo de cálculo (o computacional) relativamente corto.

4.1.2.1. Mallado en la proximidad a la pared: *inflation layer*

Para un buen mallado en la proximidad de un flujo turbulento con una pared es necesario entender lo que sucede en esta zona. Es necesario este conocimiento para cuando los valores que se están buscando son claramente dependientes de la interacción del flujo con la pared. Ejemplos son el perfil de velocidades en las regiones más cercanas, presiones o fuerzas ejercidas sobre la pared, o cualquier valor dependiente de estos.

En flujos turbulentos con números de Reynolds elevados, las tensiones presentadas en ellos son mayoritariamente turbulentas. Un modelo de turbulencias de altos números de Reynolds está pensado para trabajar en estas zonas. Sin embargo, cerca de la pared las turbulencias varían, y deben ser descritas con precisión para obtener valores correctos de variables que se ven afectadas por las condiciones del flujo en esta región. Los modelos descritos anteriormente no son capaces de realizar esta solución. Para ello, es necesario o bien el uso de diferentes modelos (de bajo Re) o la adición al modelo existente de funciones de pared.

Las funciones de pared son ecuaciones empíricas que describen el comportamiento del fluido en la región cercana. Se asume que únicamente dependen de las propiedades del fluido de esta zona y no de las alejadas.

Para determinar estas funciones, es necesario un análisis dimensional, en el que se definen la distancia a la pared (y) y la velocidad del fluido (u) de forma adimensional como y^+ y u^+ respectivamente (Ecuación 9 y Ecuación 10).

$$y^+ = \frac{y}{\nu} \left(\frac{\tau_w}{\rho} \right)^{\frac{1}{2}}$$

Ecuación 9: Altura adimensional (y^+) [5]

$$u^+ = \bar{u} \left(\frac{\tau_w}{\rho} \right)^{-\frac{1}{2}}$$

Ecuación 10: Velocidad adimensional (u^+) [5]

La relación entre estas dos magnitudes adimensionales (y^+ y u^+), es decir, el perfil de velocidad en función de la altura, depende de las tensiones en la pared (τ_w). ρ es densidad y ν es viscosidad cinemática. En el entorno más cercano a la pared se pueden distinguir diferentes zonas, descritas por ecuaciones empíricas (Figura 3):

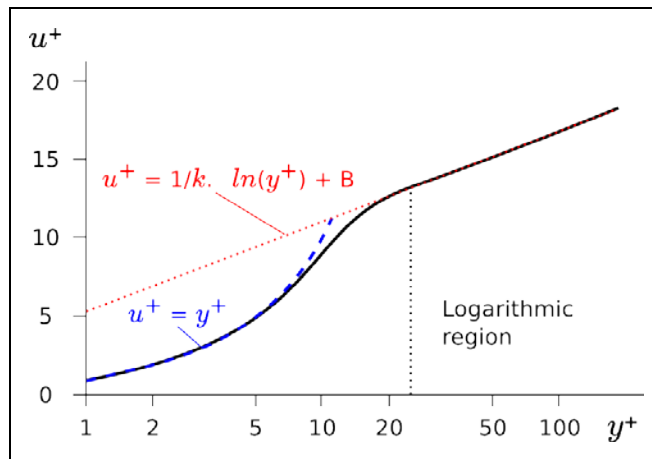


Figura 3: Gráfico con funciones de pared (rojo y azul) modelizando el comportamiento del fluido (negro) [2]

- La más cercana a la pared, la subcapa viscosa (hasta $y^+=5$). Se caracteriza porque, al estar cerca de la pared no permite la existencia de turbulencias, por lo que las tensiones que existen son prácticamente laminares, que hacen que la u^+ dependa linealmente de la y^+ (Ecuación 11).

$$u^+ = y^+$$

Ecuación 11: Ecuación de la subcapa viscosa [5]

- En una región más externa, la capa logarítmica (a partir de $y^+=30$). Las tensiones turbulentas predominan sobre las laminares, y la dependencia de la velocidad respecto la distancia con la pared es logarítmica (Ecuación 12).

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B$$

Ecuación 12: Ecuación de la capa logarítmica. [5]

Donde:

- B: constante adimensional, aproximadamente 5,1
- κ : constante de Von Karman, aproximadamente 0,41

- Zona de transición ($5 < y^+ < 30$). No se puede determinar de forma precisa el comportamiento del fluido entre las dos zonas anteriores. Las turbulencias van aumentando a medida que aumenta la distancia con la pared, por lo que las tensiones turbulentas cogen importancia.

Estas funciones son válidas exclusivamente para cuando no hay desprendimiento de capa límite.

Para poder ver estas zonas, es necesario que el primer elemento de la malla (que su base está en contacto con la pared del cuerpo) tenga una altura que lo sitúe en la zona logarítmica (como mínimo) con el fin de poder abarcar el estudio de las zonas turbulentas que afectan a estas variables que se buscan.

La altura del primer elemento, entre otros parámetros viene determinada por la y^+ que se escoja. Este mismo valor se escoge según la menor o mayor resolución requerida por el usuario y por el modelo de turbulencias usado. Para el cómputo de la altura a partir de una y^+ determinada un método es el siguiente (usado para el segundo estudio):

1. Cálculo del número de Reynolds (Ecuación 4). Se utiliza como velocidad la del flujo fuera de la región cercana a la pared.
2. Se estima el coeficiente de fricción de manera aproximada con fórmulas experimentales, por ejemplo, la siguiente (Ecuación 13):

$$c_f = \frac{0.027}{Re^{\frac{1}{4}}}$$

Ecuación 13: Aproximación de coeficiente de fricción local en placa plana mediante la ecuación de Prandtl para flujos turbulentos

3. Se calcula la tensión de la pared (Ecuación 14) mediante la fórmula que define el coeficiente de fricción:

$$c_f = \frac{\tau_w}{\frac{1}{2}\rho V_\infty^2} \rightarrow \tau_w = c_f \frac{1}{2}\rho V_\infty^2$$

Ecuación 14: Definición de coeficiente de fricción y aplicación para cálculo de la tensión en la pared

4. Se calcula la altura del primer elemento reformulando la ecuación de la y^+ (Ecuación 15):

$$y = y^+ v \left(\frac{\rho}{\tau_w} \right)^{\frac{1}{2}}$$

Ecuación 15: Reformulación de la ecuación de definición de altura adimensional

4.1.3. **Set-up**

El *set-up*, o preparación de la solución, es el estado donde las diferentes condiciones e hipótesis se introducen en el problema para acceder al resultado.

En este momento también se deciden parámetros de la simulación como:

- Propiedades del fluido (o fluidos): Se introducen parámetros como la densidad del fluido y su viscosidad.
- Propiedades del sólido que interacciona o encierra al fluido, como su rugosidad de pared.
- Condiciones de contorno: Hacen normalmente referencia a alguna de las propiedades de la entrada o salida del fluido (en caso de un problema en espacio abierto), de generación de paredes fijas, móviles, que roten o de simetrías, entre otras. La generación de simetrías es un parámetro fundamental para la disminución del tiempo computacional. También se pueden controlar otros parámetros como la gravedad en toda la simulación.
- Selección de la simulación en estacionario o en transitorio: El estado transitorio es aquél en que los resultados evolucionan en función del tiempo. Al estabilizarse la solución y ya no depender del tiempo, se considera haber llegado al estado estacionario.
- Hipótesis aplicadas a la solución: Se pueden controlar opciones como el uso de la ecuación de la energía para su resolución (que no se usen implica que en el estudio no hay transferencia de calor) o la posible compresibilidad del mismo (si $M > 0,3$). Para casos donde hay presencia de flujo turbulento, la modelización de la turbulencia es determinada por las ecuaciones designadas por el usuario.

Las hipótesis asumidas en este apartado en las propiedades de los materiales, condiciones de contorno y simplificaciones a las ecuaciones fundamentales del fluido, generan errores que alejan la realidad de la simulación. Unos parámetros más realistas normalmente implican un tiempo de simulación más alto, por lo tanto, es necesario que el usuario a la hora de la simulación realice un cierto compromiso.

4.1.4. Solver

Una vez determinadas todas las condiciones para la simulación es necesario resolver el caso.

Dependiendo del tipo de problema es importante la elección del correcto solucionador:

- Los *solvers* basados en la presión (*pressure-based solvers*) sirven desde un fluido incompresible con baja velocidad hasta compresible con alta y requieren menos capacidad computacional. Necesitan menor capacidad computacional. Estos resuelven las ecuaciones de continuidad primeramente, y las otras ecuaciones que se plantean en el problema se resuelven posteriormente a partir de los valores de las anteriores. Existe la opción de resolver de forma secuencial la presión y cantidad de movimiento, o bien a la vez (PBCS o *Pressure-Based Coupled solvers*).
- Los *solvers* basados en la densidad (*Density-Based Coupled solvers* o DBCS) se usan donde hay una dependencia significativa entre la energía, densidad, cantidad de movimiento o volúmenes con coexistencia de más de un fluido (por ejemplo, en casos de combustión). Las ecuaciones de continuidad, cantidad de movimiento energía y de la mezcla se resuelven en conjunto. La presión es obtenida a partir de la ecuación de estado, y las adicionales son calculadas posteriormente.

La solución es generada a partir de unas condiciones iniciales y un algoritmo de solución. El algoritmo linealiza las ecuaciones mediante el método de los volúmenes finitos y mediante un método iterativo encuentra la solución. El algoritmo parte de un tiempo inicial y con cada iteración procede a sumar un incremento de tiempo, en el que las variables a resolver cambian de valor, se pasa de un estado a otro.

Los algoritmos pueden ser implícitos o explícitos. Los métodos explícitos ofrecen una solución del futuro estado explícitamente a partir de una función del estado actual (Ecuación 16). Por otra parte, los implícitos contienen la solución del estado próximo en una función dependiente del estado actual y el futuro (Ecuación 17). A términos de uso, los explícitos requieren un coste computacional menor, pero necesitan un intervalo de tiempo entre estados más pequeño, ya que, si no, la solución no alcanza el los valores finales deseados (la solución no converge).

$$E(t + \Delta t) = f(E(t))$$

Ecuación 16: Ecuación a resolver en el método explícito

$$f(E(t + \Delta t), E(t)) = 0$$

Ecuación 17: Ecuación a resolver en el método implícito

Para la elección del intervalo de tiempo entre iteraciones (*timestep*), es común utilizar como indicador la Condición de Courant-Friedrichs-Lewy (Ecuación 18). En ella se especifica que para la convergencia de la solución de un algoritmo el incremento de tiempo tiene ser un número que, al relacionarlo con el tiempo que tarda el fluido a atravesar el intervalo de distancia marcado, no supere el número de Courant máximo. El número de Courant viene marcado por el tipo de algoritmo de solución escogido. Si es explícito, el Courant máximo es uno. Si es implícito puede tomar valores más altos.

$$C = \frac{\Delta t}{\frac{\Delta x}{V_x}} \leq C_{max}$$

Ecuación 18: Condición de Courant-Friedrichs-Lewy

La manera para controlar el procedimiento de la solución es la monitorización de los residuos y de variables en puntos de interés. El criterio de convergencia de la solución (para determinar que se ha llegado al estado que se quiere estudiar) normalmente reside en observar a los residuos permanecer en valores muy bajos durante un número de iteraciones significativo y los parámetros de estudio estabilizarse en valores coherentes.

4.1.5. Post-procesado

En el momento que la solución ha convergido, se obtienen los valores numéricos para cada uno de los elementos. El post-procesado tiene como función utilizar estos resultados para realizar gráficos, mapas de color, líneas de corriente, representación vectorial de variables y obtención numérica o gráfica de variables dependientes de los valores conseguidos.

En esta fase el usuario puede analizar los resultados de manera visual para poder verificar o corregir el problema por el cual se ha realizado la simulación.

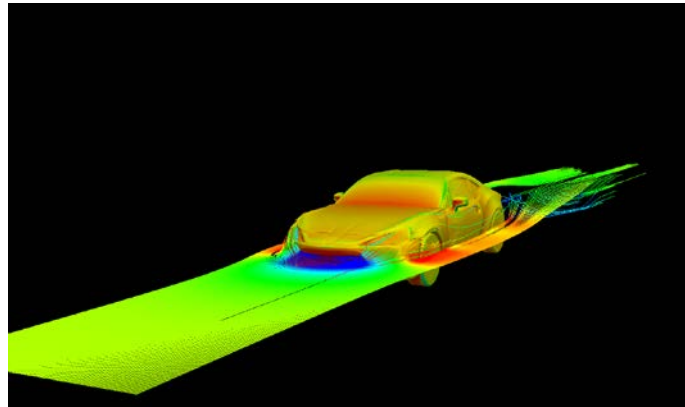


Figura 4: Líneas de corriente y mapa de color de presión sobre la superficie de un automóvil generadas en el post-procesado [6]

5. Software utilizado

El *software* utilizado para la elaboración de las simulaciones es el descrito en la (Tabla 1). Cada uno de los programas está instalado en el sistema operativo para el cual está pensado. La distribución en el caso de Linux no es exclusiva, hay más que pueden ser válidas. Aunque existía la posibilidad de haber instalado todo en el sistema operativo que ya estaba instalado en el ordenador por defecto (Windows), se ha buscado utilizar software libre también en este aspecto.

Programa	Tipo	Sistema Operativo	Uso para el primer problema	Uso para el segundo problema
ANSYS (Fluent y CFX)	Comercial	Windows	Geometría, mallado, set-up, solución, post-procesado	Geometría, mallado, set-up, solución, post-procesado
SALOME	Libre	Linux (Xubuntu)	-	Geometría y mallado
OpenFOAM	Libre	Linux (Xubuntu)	Geometría, Mallado, Set-up, Solución	Set-up, solución
Paraview	Libre	Linux (Xubuntu)	Post-procesado	Post-procesado

Tabla 1: Programas utilizados

En lo que respecta al contenido de los programas, ANSYS es un software que engloba diferentes herramientas de simulación, no exclusivamente para CFD. La interfaz de Workbench está organizada en módulos donde el usuario puede realizar cualquier parte de la simulación que le interese e interrelacionarlo con otros módulos, como hacer un mallado en uno y usarlo en diferentes módulos para hacer simulaciones diferentes (de estructuras, de transferencia de calor y de dinámica de fluidos, por ejemplo).

OpenFOAM también es un software de simulación donde se puede realizar todos los pasos para resolver una simulación numérica. La contrapartida que tiene es que no dispone de interfaz de usuario, funciona sobre comandos en el terminal y archivos escritos. Por ello, para facilitar su uso se suele utilizar SALOME para la elaboración de la geometría y el mallado, y Paraview para la visualización de los resultados para el Post-procesado. En el primer estudio se malla directamente en OpenFOAM porque la sencillez de la geometría permite que no sea un proceso muy complicado.

6. Guía de instalación de los programas necesarios para el estudio

El primer paso para realizar una simulación es disponer de los programas necesarios (los explicados en el punto anterior).

6.1.1. Instalación del sistema operativo

El archivo instalador del sistema operativo Xubuntu se puede descargar de forma gratuita a partir de la página web oficial [0](#). El método utilizado para la instalación es el uso de un dispositivo de almacenamiento externo (USB).

Para introducir el archivo en el USB es necesario un programa adicional. El recomendado por Linux es Rufus. Se puede instalar este programa sin coste desde la página web oficial. Una vez está instalado, se conecta el USB en el ordenador. Al abrir Rufus, se selecciona el dispositivo como dispositivo de destino. Se indica que el tipo de arranque es FreeDOS, y se dejan las selecciones predeterminadas para el esquema de partición (MBR) y el tipo de sistema (BIOS). Se indica la carpeta donde está el archivo de instalación descargado y se procede a empezar el proceso de escritura del USB.

Una vez el USB contiene el instalador, se apaga el ordenador, y con el mismo apagado, se introduce el dispositivo y se enciende. Muchos ordenadores viene como predeterminado que se arranque el sistema operativo prioritariamente de un dispositivo de almacenamiento externo. Si no es el caso, y accede a Windows directamente, se debe intentar otra vez el proceso, pero en el momento de encendido del ordenador se pulsa la tecla correcta para ir al menú del BIOS (software que se encarga de tener la información necesaria para acceder a cualquier sistema operativo en el ordenador). La tecla correcta depende del modelo del ordenador, pero normalmente suelen ser F1, F2, Del, Esc, F10, F11 o F12. En el menú del BIOS se selecciona como primera prioridad el arranque del sistema operativo desde el USB, y al salir del menú lo realiza.

En el momento que arranca la instalación del sistema operativo el propio proceso pide paso por paso las opciones necesarias para ello. Primeramente el instalador da a escoger entre instalar y probar Xubuntu. Se selecciona instalar. En el caso de la continuar en la instalación, si en algún punto posterior el usuario la cancela se acaba llegando al menú de prueba, dentro del cual está disponible la opción de instalarlo. Las opciones que se presentan a continuación son el idioma, opciones de instalación de actualizaciones y softwares comerciales adicionales, y el tipo de instalación. Es interesante saber que hay la opción de borrar el sistema operativo antiguo o hacer una partición al disco.

Posteriormente se selecciona la zona horaria, el idioma del teclado y el perfil de usuario y contraseña.

Una vez terminado el sistema operativo está listo para usarse. Por último, es recomendable configurar Internet (en Conexiones de Red) para la instalación de los programas.

6.1.2. Instalación de SALOME

Salome viene incluido en el paquete SALOME-Meca, junto a Code_Aster (un simulador termomecánico que no se usará en este trabajo). Para la instalación en Linux se debe descargar el archivo de la página web oficial de Code_Aster [8]. Una vez descargado, se abre el terminal y se introduce el siguiente texto:

```
tar xvf Salome-Meca-2017.0.2-LGPL-2.tgz && ./Salome-Meca-2017.0.2-LGPL-2.run
```

Este comando hará que se instale SALOME en el directorio deseado y creará un acceso directo en el escritorio.

6.1.3. Instalación de OpenFOAM y Paraview

OpenFOAM y Paraview se pueden descargar de forma conjunta mediante la página web oficial de OpenFOAM [9]. La opción de instalación a elegir es Ubuntu. Para instalarlo hay que instalar previamente la versión de Docker de Ubuntu, que distribuye versiones de OpenFOAM precompiladas.

Una vez instalado, se deben descargar los scripts mencionados en la página y ponerlos en un directorio local. Después de eso, se ejecutan los siguientes comandos en este orden:

```
chmod +x installOpenFOAM
```

```
./installOpenFOAM
```

```
chmod +x startOpenFOAM
```

```
./startOpenFOAM
```

El último abrirá un entorno en el que está listo para usarse OpenFOAM.

7. Guía de uso para estudios de flujo interno y externo

Una vez planteados los conceptos relacionados con la simulación y su software, se realizan ambos estudios. El primer problema planteado, debido a su simplicidad, la geometría y el mallado se realizan en OpenFOAM y el post-procesado visualizado en Paraview. En el segundo caso se realizarán los dos primeros pasos mediante SALOME. Ambos estudios en la extracción de resultados y conclusiones se comparará con los resultados obtenidos por ANSYS Fluent (en el primer caso) o CFX (segundo caso) del mismo problema.

Por lo general, los casos que se estudian en OpenFOAM contienen estas carpetas principales:

- Carpetas de tiempo: Si la simulación no se ha solucionado, únicamente contiene la carpeta 0. Estos ficheros poseen la información de las variables de estudio. En el caso de la carpeta 0 (si se ha seleccionado como inicialización de la solución en el tiempo 0) representa el valor inicial de las variables en el problema. La información de variables se suele dar en unidades en sistema internacional con un campo de todas las dimensiones registradas en OpenFOAM. Las dimensiones (Figura 5) están representadas en el siguiente orden: masa (kg), posición o distancia (m), tiempo (s), temperatura (K), cantidad (mol), intensidad de corriente (A) e intensidad lumínica (cd).

```
dimensions      [0 1 -1 0 0 0 0]
```

Figura 5: Representación de las dimensiones para la velocidad (m/s)

- Carpeta *constant*: Contiene toda la descripción de la geometría mallada (subcarpeta *polyMesh*) y las propiedades físicas del fluido aplicadas (como el modelo de turbulencias o la viscosidad cinemática, por ejemplo).
- Carpeta *system*: Abarca la información del solucionador. Sus tres ficheros principales son los siguientes:
 - o *controlDict*: Está presente la información del tipo de algoritmo de solución, inicio y fin de las iteraciones, el incremento de tiempo y las opciones para la salida de datos.

- *fvSchemes*: Comprende los parámetros de discretización de las ecuaciones que se usan durante el proceso de solución.
- *fvSolution*: Es el fichero donde se encuentran los algoritmos de solución de las ecuaciones con sus opciones y tolerancias.

Estas carpetas son las que se necesitan modificar (con las condiciones adecuadas) para poder obtener una solución correcta. En ambos casos de estudio se explicará el procedimiento a seguir para ello.

7.1. Estudio comparativo de flujo interno

La simulación se basará en un recipiente cerrado completamente lleno de agua (Figura 6) en el que la pared superior (simulando una cinta) es movida a una velocidad determinada, haciendo que se mueva todo el líquido del interior. Aunque el estudio pretende ser bidimensional, es necesaria la aplicación al volumen de la coordenada Z, ya que *OpenFOAM* no permite realizar estudios que no sean 3D.

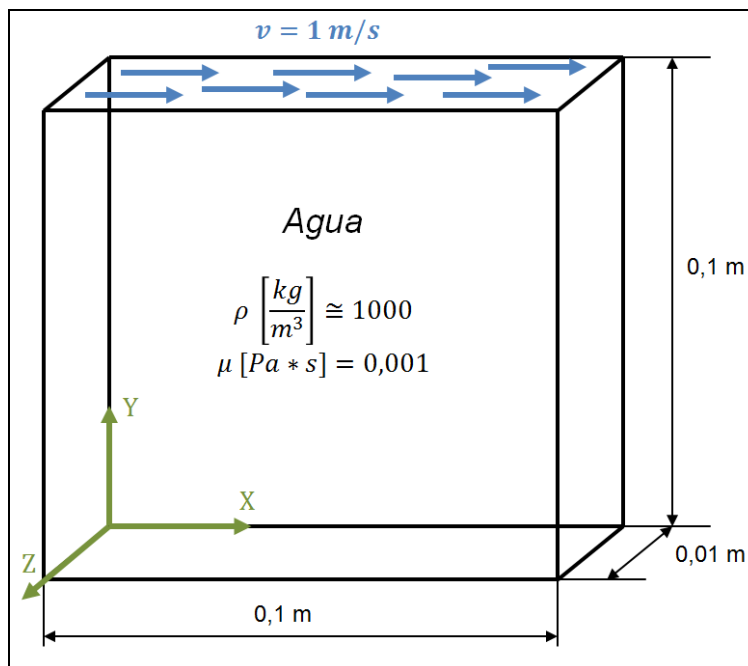


Figura 6: Geometría, condiciones y propiedades del estudio del fluido en un entorno cerrado

7.1.1. Geometría y mallado

La geometría consiste en un volumen de 0,1 m de altura y base, y 0,01 de grosor (el grosor es irrelevante, es un estudio en 2D).

El volumen es mallado directamente por comandos en un documento para OpenFOAM. Se describen en este los vértices del hexaedro y las divisiones para el mallado. El mallado presenta 20 divisiones de la geometría en las coordenadas X e Y. Para la Z no es necesaria la división. Tiene un total de 400 elementos. No es necesario un estudio de calidad del mallado actual, dado que son elementos hexaédricos regulares. Aunque tuvieran una relación de aspecto grande del eje Z respecto a los otros no sería problema, debido a que el flujo no irá en esta dirección. En este caso el mallado no se ha enfocado a estudiar el comportamiento del fluido en las zonas cercanas a la pared, por lo que no se genera una estructura más fina en tal zona.

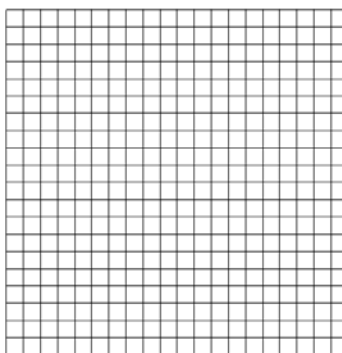


Figura 7: Mallado inicial propuesto

Para configurar estas condiciones en OpenFOAM, se parte de un tutorial que viene con la instalación del programa.

Para acceder a OpenFOAM, copiar el documento con todo el proceso del caso contenido, y abrir la carpeta de estudio se ejecutan respectivamente los siguientes comandos en el terminal:

```
cd $FOAM_RUN
```

```
cp -r $FOAM_TUTORIALS/incompressible/pisoFoam/ras/cavity .
```

```
cd cavity
```

Una vez dentro del directorio, la información del mallado se encuentra contenida en *blockMeshDict*, dentro de *system*. La información que contiene es la descrita en la Figura 8 y la Figura 9. La línea 17 contiene un conversor para la información de los puntos siguientes descritos multiplicarlos por el factor indicado. En las siguientes líneas se describen las coordenadas de los vértices de la geometría. En *blocks*, se describen las regiones de mallado (en este caso una) con el comando que divide la región en los elementos según indicado en las opciones (20 en dirección X, 20 en dirección Y y 1 en Z).

Para un estudio de sensibilidad de mallado modificando estos números varía el número total de elementos.

En *boundary* se describen los nombres y el tratamiento que tienen las superficies que engloban el problema. Todas se categorizan como *wall*, excepto de *frontAndBack*, ya que con la condición *empty* se consigue que esta superficie no afecte al estudio (hace que el problema sea bidimensional).

```
1 /*-----*-- C++ --*-----*/
2 |-----|
3 | \ /   F i e l d       | O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x
4 | \ /   O p e r a t i o n   | V e r s i o n :   4 . 1
5 | \ /   A n d             | W e b :   w w w . O p e n F O A M . o r g
6 | \ /   M a n i p u l a t i o n |
7 |-----*--*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       blockMeshDict;
14 }
15 // ***** //
16
17 convertToMeters 0.1;
18
19 vertices
20 (
21     (0 0 0)
22     (1 0 0)
23     (1 1 0)
24     (0 1 0)
25     (0 0 0.1)
26     (1 0 0.1)
27     (1 1 0.1)
28     (0 1 0.1)
29 );
30
31 blocks
32 (
33     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34 );
35
36 edges
37 (
38 );
```

Figura 8: Información del mallado en el fichero blockMeshDict (1/2)


```

35
36 edges
37 (
38 );
39
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;
53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
70
71 mergePatchPairs
72 (
73 );
74
75 // *****

```

Figura 9: Información del mallado en el fichero *blockMeshDict* (2/2)

Para ejecutar el mallado en el terminal, en el directorio del caso de estudio, se ejecuta el siguiente comando:

blockMesh

Este comando genera la carpeta *polyMesh* con toda la información del mallado ejecutada.

7.1.2. Set-up y solución

Las condiciones de contorno del problema son las siguientes:

- Pared superior con movimiento en eje X a velocidad de 1m/s
- A las paredes frontal y trasera se les aplica la condición *empty*.
- Resto de paredes estáticas, sin deslizamiento (*noSlip*).

Para la aplicación de estas condiciones de contorno es necesario modificar el archivo U de la carpeta 0 poniendo dichos valores de la siguiente manera (Figura 10):

```

1 /*-----*- C++ -*-----*/
2 |=====|
3 | \ \ \ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
4 | \ \ \ / | O p e r a t i o n | Version: 4.1
5 | \ \ \ / | A n d | Web: www.OpenFOAM.org
6 | \ \ \ / | M a n i p u l a t i o n |
7 /*-----*- C++ -*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     object        U;
14 }
15 // ***** //
16
17 dimensions      [0 1 -1 0 0 0 0];
18
19 internalField   uniform (0 0 0);
20
21 boundaryField
22 {
23     movingWall
24     {
25         type      fixedValue;
26         value     uniform (1 0 0);
27     }
28
29     fixedWalls
30     {
31         type      noSlip;
32     }
33
34     frontAndBack
35     {
36         type      empty;
37     }
38 }
39
40 // ***** //
41

```

Figura 10: Condiciones aplicadas en el archivo U

Las propiedades del agua usadas en este estudio son:

- Densidad = 1000 kg/m^3
- Viscosidad dinámica = $0,001 \text{ Pa}\cdot\text{s}$

OpenFOAM trabaja directamente con viscosidad cinemática (el valor resultado de dividir la viscosidad dinámica entre la densidad). Se modifica en el fichero *transportProperties* de la

carpeta *constant*.

Las hipótesis en las que se basará la simulación son:

- Simulación en régimen estacionario.
- El flujo se considera incompresible (no se tiene en cuenta la posible variación de densidad del fluido), ya que al tratar con esta magnitud de velocidad el Número Mach (M) es inferior a 0,3.
- No se considera el efecto de la temperatura, se supone una temperatura constante en todo el volumen de 25°C.
- La presión de referencia es de 101325 Pa (atmosférica).
- El flujo es turbulento al tener un $Re=100000$ (calculado a partir de la Ecuación 4), donde la longitud característica usada es el ancho del volumen, 0,1m). El modelo de turbulencias usado para modelar el comportamiento del fluido en ambos programas es el de $k-\epsilon$. Se ha escogido un modelo de ecuaciones RANS (ecuaciones *Reynolds-averaged Navier-Stokes*), y en concreto éste para agilizar el tiempo de cálculo y por ser uno de los modelos más usados en la actualidad para simulaciones de con una aceptable resolución de resultados. Requiere poca capacidad computacional y es más fácil que otros modelos que converja. Es un modelo para altos números de Reynolds, por lo que requiere unas funciones de pared. Se aplican funciones de pared estándar.

Al no tener una dependencia entre las variables de estudio, el tipo de solucionador escogido es *Pressure-Based solver*. Para el algoritmo de solución de velocidad y presión se utiliza el PISO (*Pressure-Implicit with Splitting of Operators*). El motivo de elección es que en la actualidad es comúnmente utilizado para flujos estacionarios.

Tanto el modelo de turbulencias como el tipo de solucionador vienen dados por el tutorial, por lo que no es necesario modificarlos. Es recomendable a la hora de realizar un problema sencillo buscar un tutorial con los parámetros del solucionador y modelo de turbulencias deseado. En toda la biblioteca de tutoriales de OpenFOAM hay para un gran número de casos y así no es necesario introducir cada valor manualmente.

La determinación del timestep en OpenFOAM sigue la indicación de la Condición de Courant-Friedrichs-Lewy (Ecuación 18). Se decide para un caso extremo, al ser un algoritmo que usa un método implícito número de Courant máximo de 2 (más grande que 1 para llegar la solución en menor tiempo pero sin hacer que el resultado no converja). Como velocidad se utiliza la de la pared (1m/s) y el incremento de espacio el ancho de celda

(0,005 m). Se obtiene un incremento de tiempo de 0.01s.

La situación inicial del problema se determina con todo el volumen a velocidad nula y la presión atmosférica (en *internalFields* de los ficheros de u y p de la carpeta 0 se indica con valor 0).

Se realizan un número 2000 iteraciones de forma que queda de forma clara que se ha llegado a una convergencia de la solución. Estos parámetros se introducen en el fichero *controlDict* de la carpeta *system*. Se ve una reducción constante de los residuos de las ecuaciones a resolver en ambos programas hasta un punto donde prácticamente no hay error.

```
18 application      pisoFoam;
19
20 startFrom        startTime;
21
22 startTime        0;
23
24 stopAt           endTime;
25
26 endTime          20;
27
28 deltaT           0.010;
29
30 writeControl     timeStep;
31
32 writeInterval    1;
33
34 purgeWrite       0;
35
36 writeFormat      ascii;
37
38 writePrecision   6;
39
40 writeCompression off;
41
42 timeFormat       general;
43
44 timePrecision    6;
45
46 runTimeModifiable true;
```

Figura 11: Parámetros en el fichero *controlDict*

Para realizar la solución basta con ejecutar el comando *pisoFoam* en el terminal localizado en el directorio del caso de estudio, y mientras se ejecuta se irán representando los residuos en directo en pantalla. Se puede guardar el registro en un archivo si en vez de usar el comando anterior se utiliza *pisoFoam > log*.

7.1.3. Post-procesado, resultados y conclusiones

Para acceder a *Paraview*, una vez terminada la solución se ejecuta en el terminal el comando:

```
paraFoam &
```

Para poder ver una superficie o volumen cualquiera en el menú de propiedades, se puede modificar la representación y el color de lo representado, ya sea monocolor o una escala de colores que dependen de una variable. Así se pueden crear los mapas de color de presión y velocidad.

Para corroborar la convergencia de la solución se decide monitorizar una serie de puntos arbitrarios. Los puntos se sitúan en la parte de la geometría indicada en la (Figura 12). Para obtener la información de estos puntos en *Paraview* se debe seleccionar en *Filter > Search > PlotSelectionOverTime* el punto de interés y guardar los datos obtenidos (que se pueden ver abriendo un nuevo *Layout* con *Spreadsheet View*) con en un archivo *.csv* (*File > Save Data*).

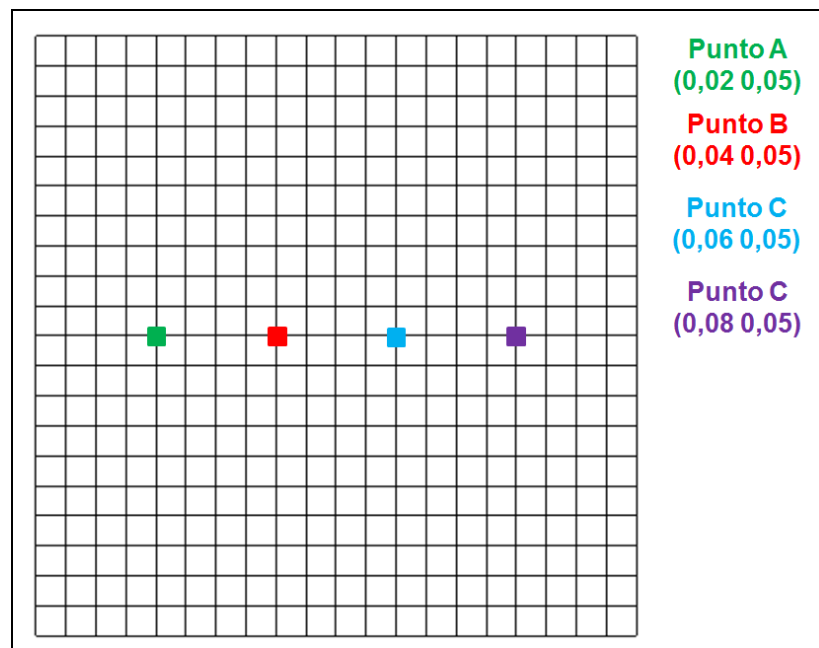


Figura 12: Puntos de monitorización con coordenadas en los ejes (X,Y)

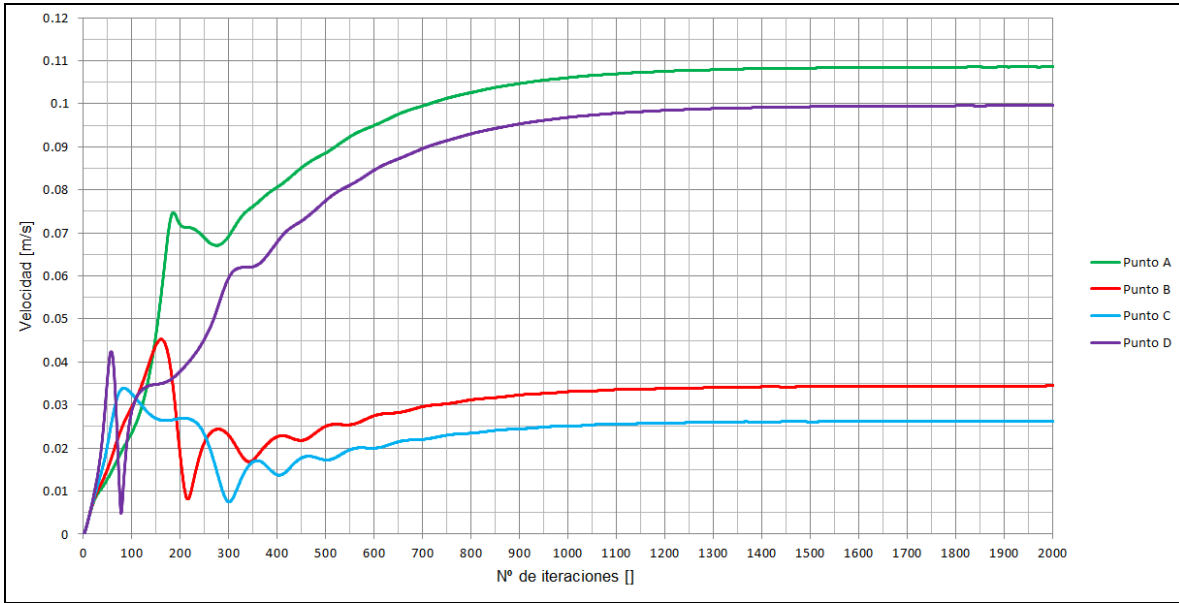


Figura 13: Monitorización de la velocidad en OpenFOAM

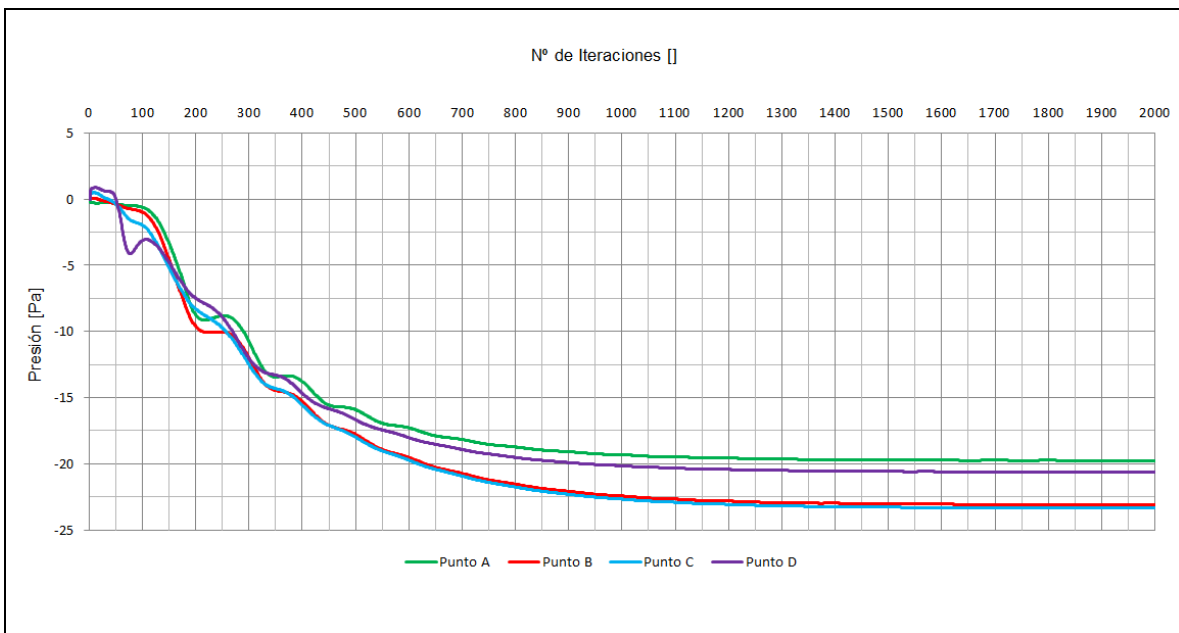


Figura 14: Monitorización de la presión en OpenFOAM

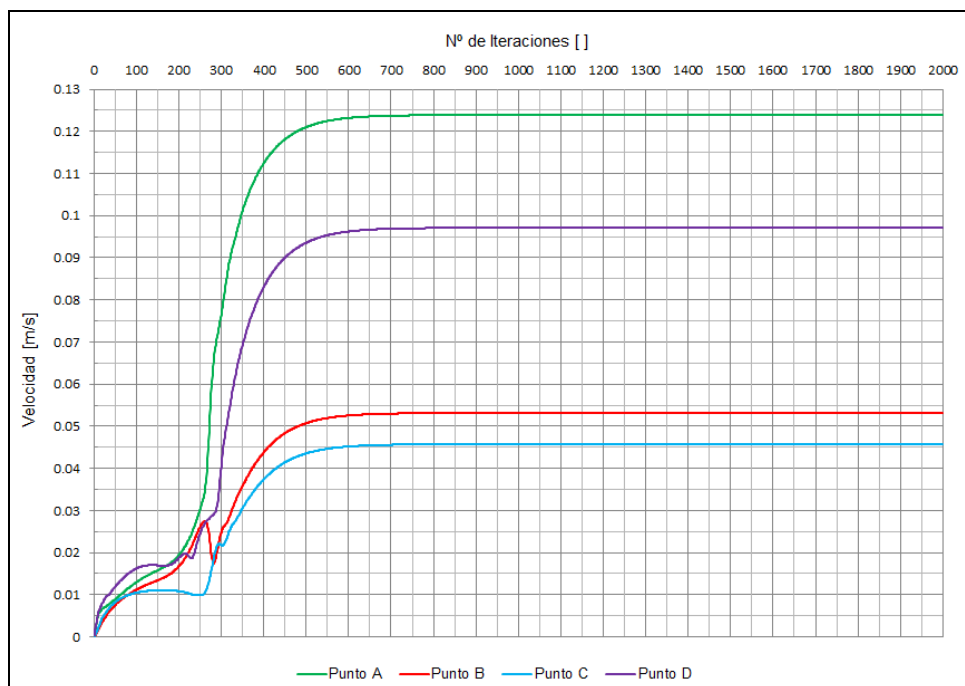


Figura 15: Monitorización de la velocidad en ANSYS

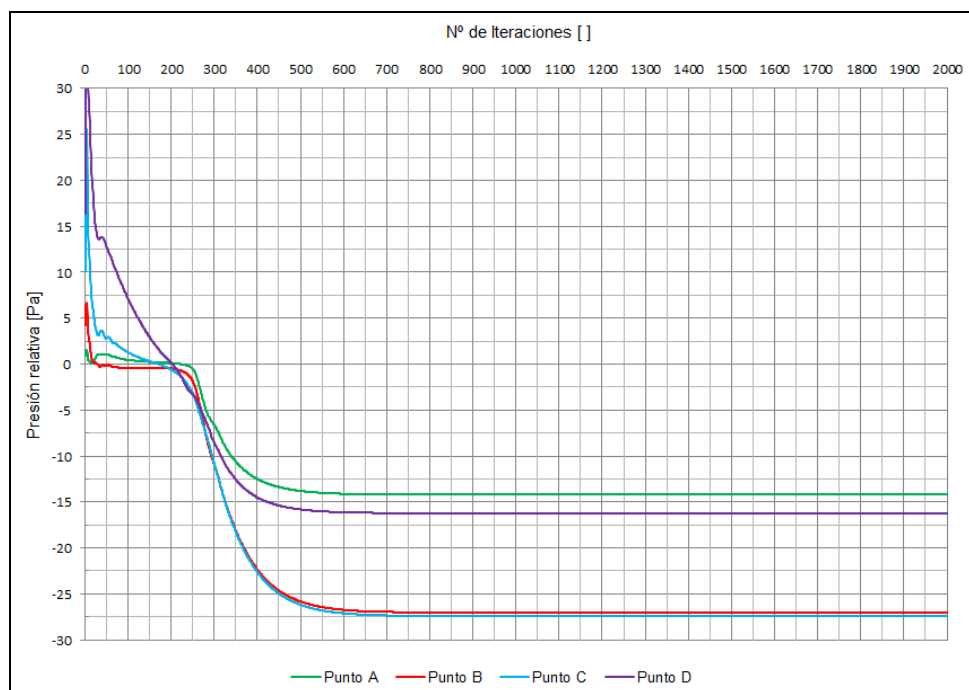


Figura 16: Monitorización de la presión en ANSYS

En la monitorización de puntos también ha sido visible tanto en OpenFOAM (Figura 13 y Figura 14) como en ANSYS (Figura 15 y Figura 16) la convergencia de los resultados (cada uno de los puntos se han estabilizado en un valor).

Una vez alcanzados los resultados con las dos herramientas de simulación, se representa de forma gráfica los vectores velocidad (en OpenFOAM mediante *Filter > Search > Glyph*) para poder hacer una primera comparación visual. Se puede observar que son unos resultados relativamente iguales (Figura 18 y Figura 17).

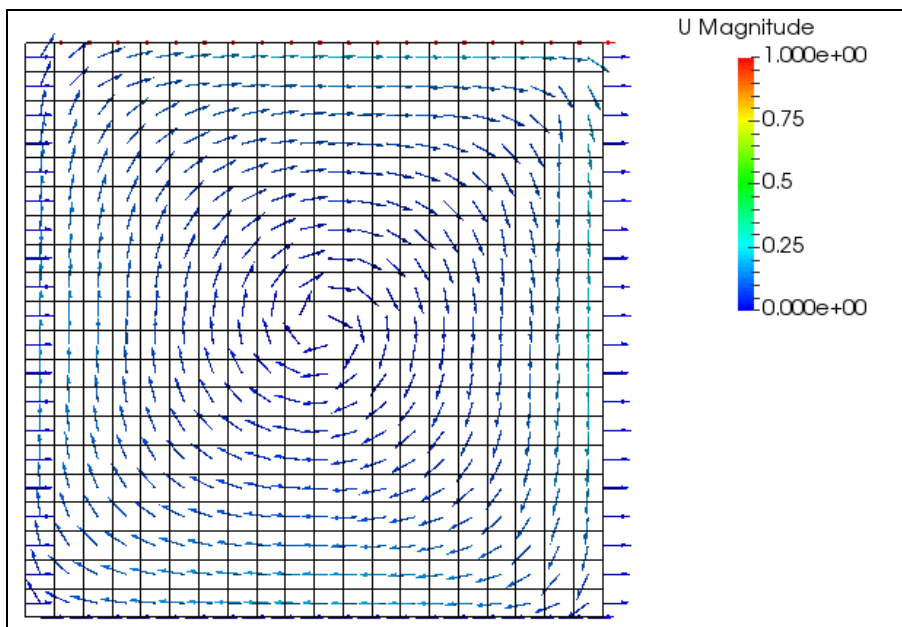


Figura 17: Vectores de velocidad representados en Paraview (solucionado por OpenFOAM).

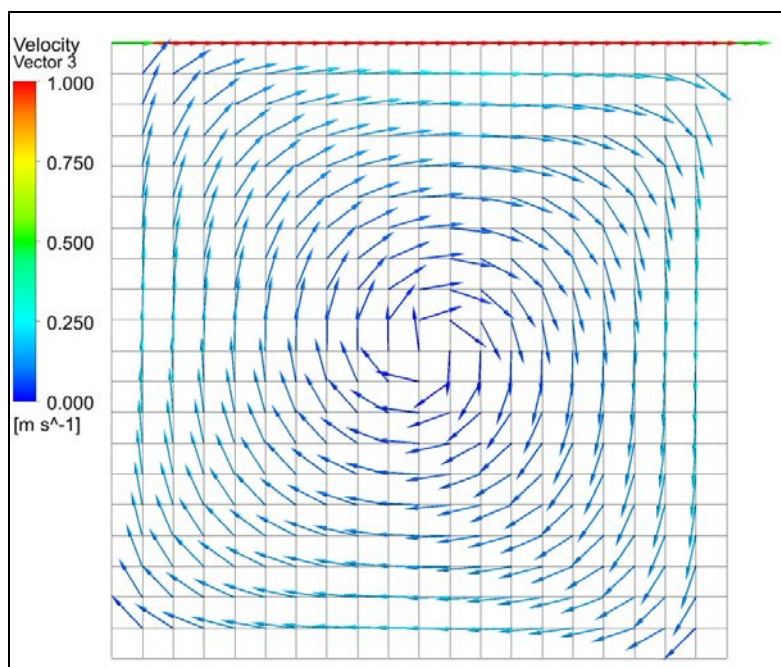


Figura 18: Vectores de velocidad representados en ANSYS Fluent.

Las diferencias en la pared de la simulación por OpenFOAM son debidas a que en

Paraview también se han representado vectores en la misma pared, que como condición aplicada, el fluido no desliza, por lo tanto su velocidad es nula. La dirección representada es arbitraria, pero el valor es nulo.

En general, se puede ver que al ser un recinto cerrado y mover mediante la pared superior el fluido en una dirección, en la zona inferior la dirección es contraria (porque no hay ni entradas ni salidas de fluido). Por este motivo se forma un remolino cuyo centro coincide aproximadamente con el del volumen, donde la velocidad es prácticamente nula.

Referente a las presiones el mapa de colores es el de la Figura 19 y la Figura 20. En OpenFOAM las presiones, al no indicar lo contrario, se representan divididas por la densidad. Para obtener el valor de presión se ha creado una variable posteriormente en el post-procesado multiplicándola. Para ello, seleccionado el caso de estudio en *Pipeline Browser*, se hace uso de la calculadora (*Filters > Search > Calculator*). Se seleccionan como datos los ofrecidos por las celdas, y se multiplica la presión (en la pestaña de escalares) por la densidad (1000 kg/m^3).

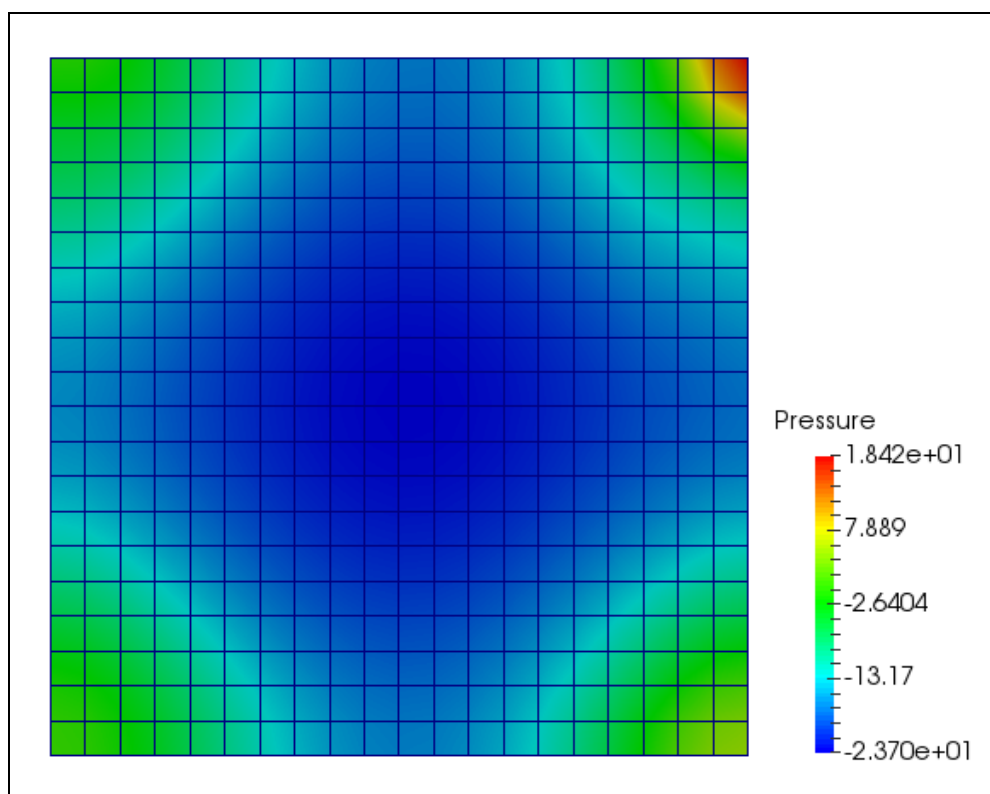


Figura 19: Mapa de presiones en Paraview (OpenFOAM)

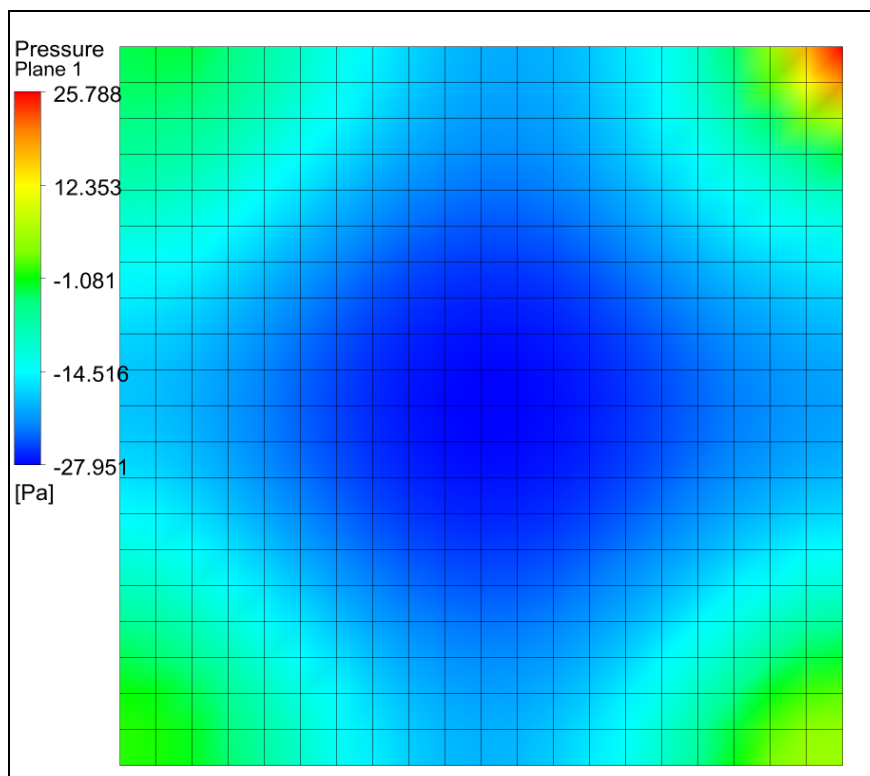


Figura 20: Mapa de presiones en ANSYS

Se puede observar que los resultados entre la Figura 19 y Figura 20 también son semejantes. También tienen concordancia con las condiciones del problema planteadas: en la esquina donde termina la pared móvil, el flujo inferior a esta impacta con la pared fija de la derecha, haciendo que se reduzca la velocidad, y por lo tanto, que aumente su presión. A esta zona se la podría denominar de estancamiento, al igual que las otras tres esquinas (aunque en menor medida). La zona central tiene unos valores de presión y velocidad bajos, que indican que es una zona turbulenta.

Para observar con detalle los datos, se selecciona una línea horizontal que atraviesa la geometría ($Y=0,05\text{m}$), y se grafican los valores del módulo de la velocidad y de la presión (Figura 21 y Figura 22). Para realizarlo en OpenFOAM, se debe seleccionar la opción `Filters > Search > Plot Over Line`, y seleccionar dos puntos extremos de la línea a graficar. Los datos se pueden guardar de la misma manera que la monitorización de los resultados.

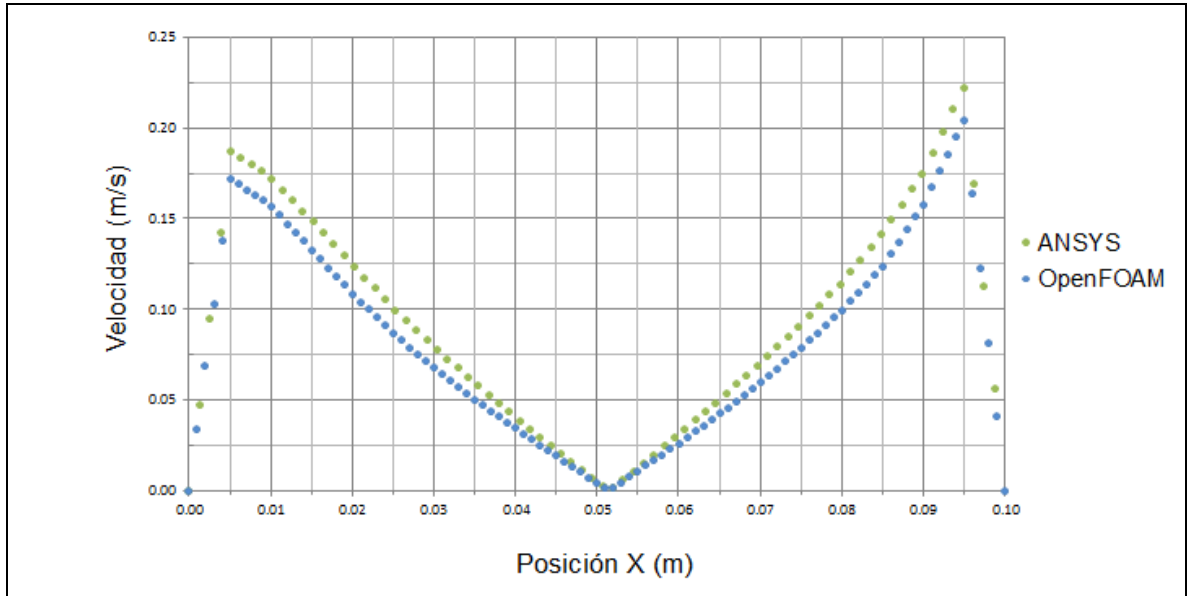


Figura 21: Gráfico comparativo de velocidades en una línea

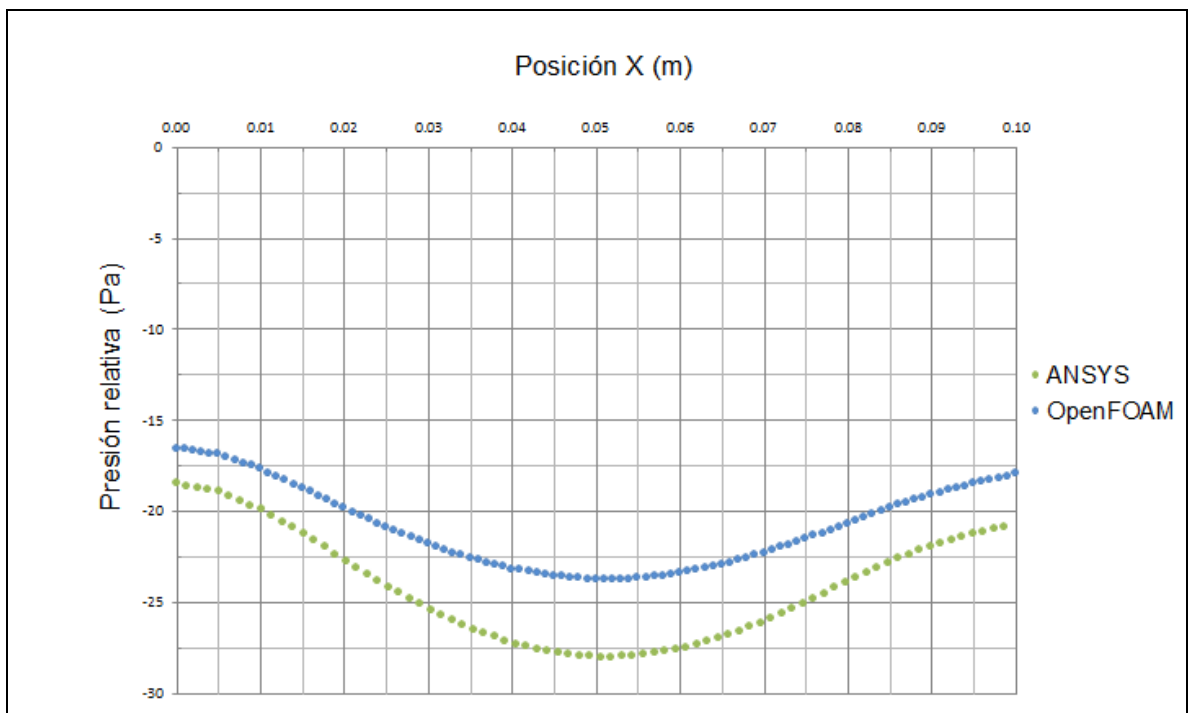


Figura 22: Gráfico comparativo de presiones en línea

Referente a la velocidad, ambos *softwares* ofrecen unos resultados con un parecido razonable. Las zonas donde su valor es más pequeño cerca de las dos paredes, a medida de acercarse, su velocidad disminuye. Esto se corresponde a la capa límite descrita al programa con la función de pared estándar. Aun así, en ninguno de los *softwares* los valores deben ser tomados en cuenta: al no determinar un mallado lo suficientemente

pequeño en la región cercana a la pared, la solución en esa región no es representativa. Además, en la zona central también se puede ver una zona de velocidades y presiones bajas, que se refleja también en la representación vectorial (Figura 17), que es consecuencia de la zona turbulenta ubicada en esa área.

Con respecto a las presiones relativas en la línea, los valores difieren un poco más, aunque son del mismo orden. La diferencia mayor es en el centro, considerada zona turbulenta (y por lo tanto con un comportamiento caótico), y su valor máximo ronda sobre los 10 Pa (no es una diferencia significativamente alta). Al ser el mallado del mismo tipo y tener el mismo modelo de turbulencias se descartan ser causas de la diferencia. El error se mantiene en toda la línea, por lo que tampoco puede haber sido causa de las funciones de pared. Considerando las opciones restantes, puede ser debido a los parámetros usados en el algoritmo para el cálculo de la solución, que se han dejado en cada uno de los programas por defecto.

7.2. Estudio comparativo de flujo externo

Una vez se realiza el estudio de fluido interno, se procede a estudiar la interacción de un fluido externo con un perfil de ala conocido. El objetivo de este estudio es corroborar los resultados obtenidos por la solución de OpenFOAM con ANSYS con un mallado más complejo, generado por el programa SALOME.

Para ello el problema planteado es de un flujo de agua que atraviesa un perfil NACA0012 [10]. El perfil se simulará inclinado un ángulo de 2° . Para simplificar el caso en la simulación, en vez de inclinar el perfil se reorienta el vector velocidad. Los valores a estudiar serán los coeficientes aerodinámicos de sustentación y de arrastre del perfil.

7.2.1. Geometría

Primeramente se copia la carpeta que sirve de base para este estudio. En este caso será la carpeta *pitzDaily* (situada en *incompressible > pimpleFoam*)

La geometría a realizar va a ser bidimensional, ya que la extrusión a 3D se realizará en el mallado.

Lo primero que se debe hacer en SALOME es abrir un caso nuevo en *File > New*. Se selecciona el módulo de geometría para poder realizar el volumen de control del caso a estudiar.

La realización de este perfil NACA es a partir de propia ecuación (Ecuación 19):

$$y = \pm \frac{t}{0,2} (A\sqrt{x} + Bx + Cx^2 + Dx^3 + Ex^4)$$

Ecuación 19: Ecuación para la obtención de perfiles de ala NACA simétrico de 4 dígitos

Donde:

- t: ratio del ancho máximo de ala y la cuerda. En este caso, como es NACA 0012 tiene un 12% de grosor máximo respecto la cuerda, que para el caso del problema será de 1m. Por lo tanto, el ratio es de 0,12.
- A,B,C,D,E: coeficientes con valor: A=0,2969; B=-0,1260; C=-0,3516; D=0,2843; E=-0,1015

Para hacer este perfil, se crea con dos curvas, una para la parte positiva y otra para la negativa (*New Entity > Basic > Curve*). En las opciones de control (Figura 23) se selecciona la tercera opción de interpolación, el modo de creación de forma analítica y los parámetros de la curva a partir de la ecuación descrita aplicada para el NACA 0012.

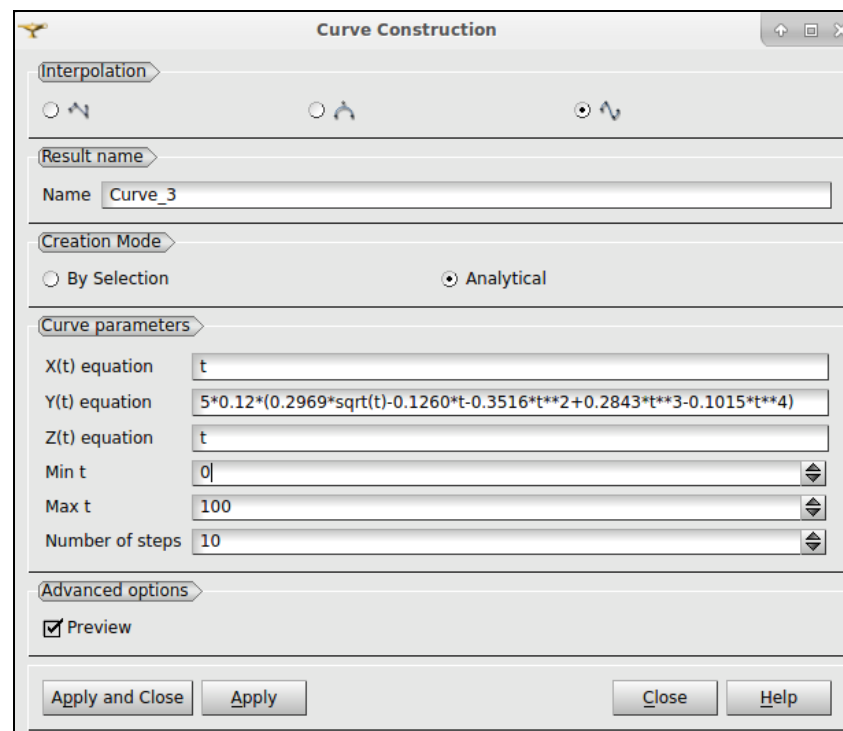


Figura 23: Opciones de control de generación de curvas en SALOME

Alternativamente se puede usar un modelo en 3D ya generado que se puede encontrar en páginas como *3D Content Central* [11].

Una vez creado el perfil, se debe unir el extremo del ala. Para ello, mediante una línea

(*New Entity > Basic > Line*), se seleccionan los dos puntos de cada extremo del perfil superior e inferior del ala y se aplica.

Después de crear el perfil completo, se debe delimitar el entorno de estudio. La geometría propuesta es la siguiente (Figura 24):

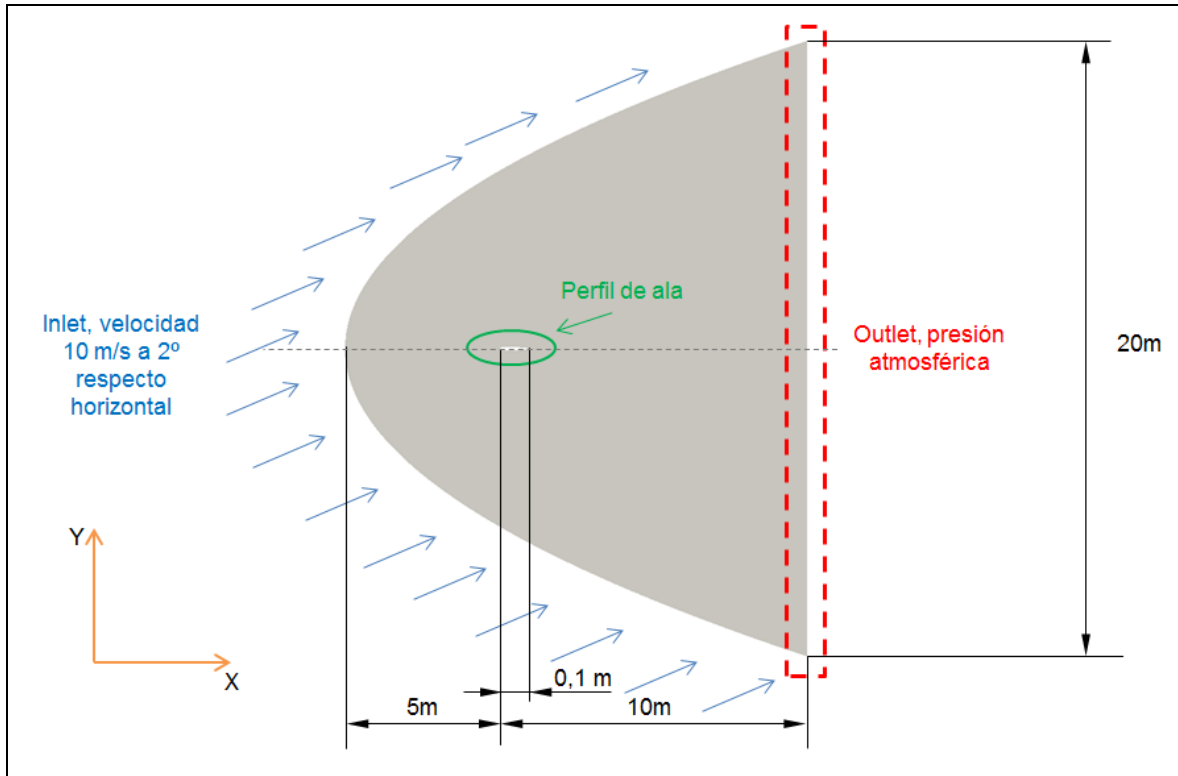


Figura 24: Geometría, condiciones y propiedades del estudio del fluido en un entorno abierto

El volumen (de momento superficie) de control es generado de manera que tanto la entrada de fluido (*inlet*) como la salida (*outlet*) no estén demasiado próximas al perfil, sin hacer un volumen excesivamente grande. El tener un espacio entre ellos permite que el desarrollo del flujo sea el correcto. Sino, es alterado y la solución se desvirtúa del valor real. El origen de coordenadas se sitúa en el extremo delantero del ala. El problema también es bidimensional, aunque se le da un grosor en el eje Z con el único fin de poder hacer la simulación.

Para ello, se crean tres puntos (*New Entity > Basic > Point*) en las coordenadas $V1=(-5,0,0)$, $V2=(10,10,0)$ y $V3=(10,-10,0)$. Después de realizarlos, se deben unir de forma que queden como en la figura (Figura 25) creando una curva (con la tercera opción de interpolación y seleccionando los tres puntos en el orden $V2$, $V1$ y $V3$) y también una recta para cerrar el conjunto.

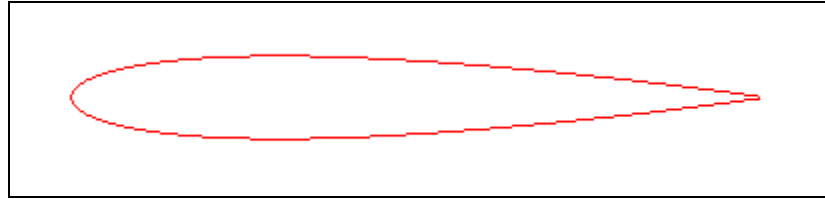


Figura 25: Perfil del ala dibujado en SALOME

Para crear la superficie que será el volumen de control, en el menú de *New Entity > Build > Face* se seleccionan todas las líneas creadas anteriormente (seleccionándolas a la vez usando el menú de buscador de objetos, a la derecha).

La geometría de esta manera está realizada, pero para prepararla para el mallado se debe hacer lo siguiente:

- *New Entity > Primitive > Disk*. Se selecciona un radio de 20 y la orientación en el eje OZX.
- *Operations > Partition*. Se escoge como objeto la cara realizada antes, y como objeto herramienta el disco. El tipo de resultado debe ser otra cara.
- *New Entity > Explode*. El objeto principal es la partición realizada antes y el tipo de subformas son líneas (*edge*).
- En el menú del buscador de objetos ahora aparecen un conjunto de líneas nuevas. Algunas de ellas servirán para las condiciones de contorno y otras para el proceso de mallado. Se crean grupos (*New Entity > Group > Create Group*) para cada uno de estos casos, utilizando como forma principal la partición. Los grupos son:
 - o Las dos líneas que forman el *inlet*, nombrándolas como “*inlet*”
 - o Las dos líneas que forman el *outlet*, nombrándolas como “*outlet*”
 - o Las dos curvas y los dos trozos de recta que forman el perfil de ala, nombrándolas como “perfil”
- Las dos líneas restantes horizontales que parten la geometría en dos se les cambia el nombre a A y B (la que está más cerca del *inlet* y la que está más cerca del *outlet*, respectivamente).
- Se vuelve a hacer *Explode* con las mismas condiciones pero se selecciona como subformas caras (*face*). De las dos caras formadas se hace un grupo, que corresponderá con la parte de detrás del volumen de control una vez extruido. Se

renombra a “back”.

7.2.2. Mallado

El diseño para el mallado de este estudio tiene como objetivo presentar una mayor definición en zonas cercanas a la interacción con el perfil de ala e ir aumentando el tamaño en las zonas lejanas de manera progresiva (para ahorrar potencia computacional total necesaria para alcanzar la solución).

Para conseguir lo planteado se ha de crear una malla bidimensional y extruirla en la tercera. Con ese objetivo, en SALOME se selecciona *Mesh > Create Mesh*. La geometría a seleccionar es la partición realizada en la geometría. El algoritmo de mallado de 2D que se escoge es Netgen-2D, con hipótesis adicional de *Viscous Layers 2D*. Esto se realiza para obtener la capa de elementos hexaédricos propia del mallado cerca de una pared (*inflation layer*). Las propiedades necesarias de esta capa se han estudiado a partir de la metodología en el apartado 4.1.2.1, con una y^+ de 200 (escogida por recomendación del modelo de turbulencias elegido):

$$Re = \frac{\rho VL}{\mu} = \frac{1000 \cdot 10 \cdot 1}{0.001} = 1 \cdot 10^7$$

$$c_f = \frac{0.027}{Re^{\frac{1}{7}}} = 0,0026$$

$$\tau_w = c_f \frac{1}{2} \rho V_\infty^2 = 130 \text{ Pa}$$

$$y = y^+ v \left(\frac{\rho}{\tau_w} \right)^{\frac{1}{2}} = 0,000554 \text{ m (con } y^+ = 200)$$

Por lo tanto, la altura del primer elemento debe ser como máximo de 0,000554 m. Para poder ver bien toda la región, se modeliza con 20 capas, que van creciendo un 120% de altura a medida que se alejan del perfil, con un ancho total de 0,1034 m. Los valores, en resumen, indicados en la hipótesis adicional son las descritas en la (Figura 26)

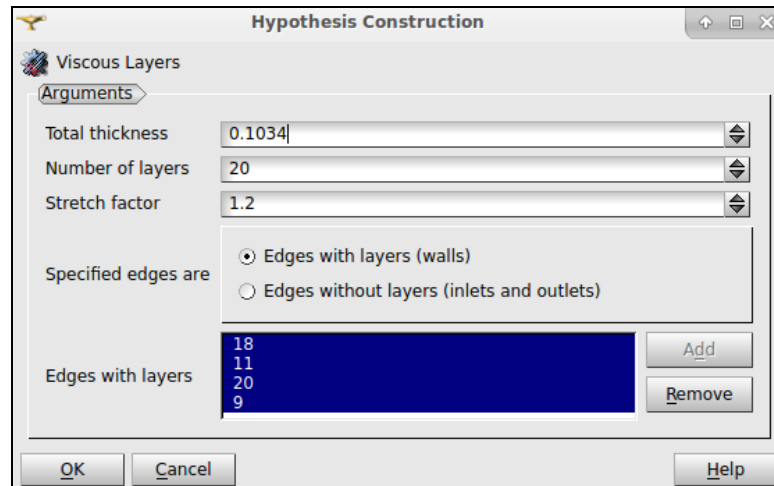


Figura 26: Valores de la hipótesis adicional

Las paredes añadidas son las correspondientes al grupo hecho en la geometría del perfil.

Para las condiciones del mallado en 1D, se selecciona como algoritmo *Wire Discretisation* y como hipótesis *Local Length* de 0,1 m (para tener un ancho máximo de ese valor).

Con el fin de crear un aumento gradual del tamaño de mallado, se crean dos condiciones de mallado (Botón derecho en *Object Browser a Mesh > Create Sub-mesh*), con geometrías A y B (líneas creadas en el módulo de geometría), tipo de algoritmo *Wire Discretization* con hipótesis de número de segmentos. En esta hipótesis en el caso de A se seleccionan 50 segmentos con un factor de escala de 100, y seleccionando como opción de *Reversed Edges* el B. Para el caso de B son 100 segmentos con un factor de escala de 200, con *Reversed Edges* B. Como hipótesis adicional para ambos se elige *Propagation of 1D Hypothesis on Opposite Edges*.

Se ejecuta el mallado con botón derecho a *Mesh* en el *Object Browser* y se selecciona *Compute*. El resultado es el siguiente (Figura 27, Figura 28y Figura 29):

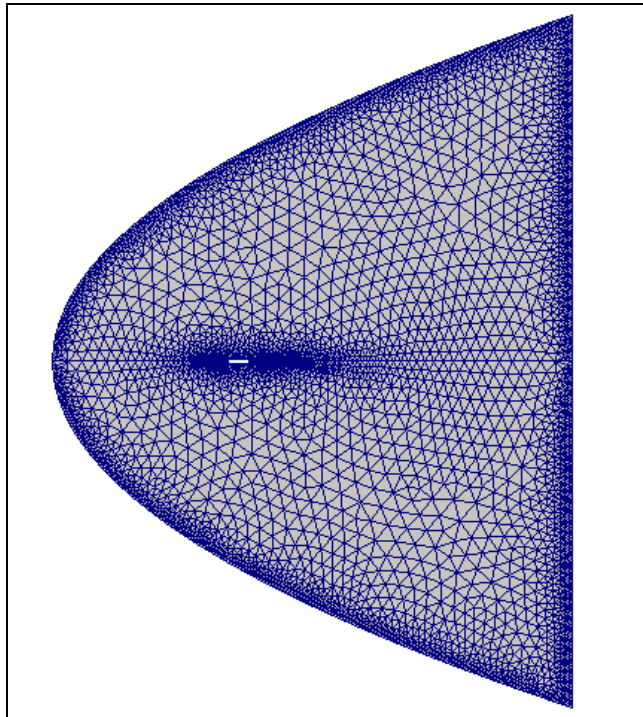


Figura 27: Mallado planteado en OpenFOAM

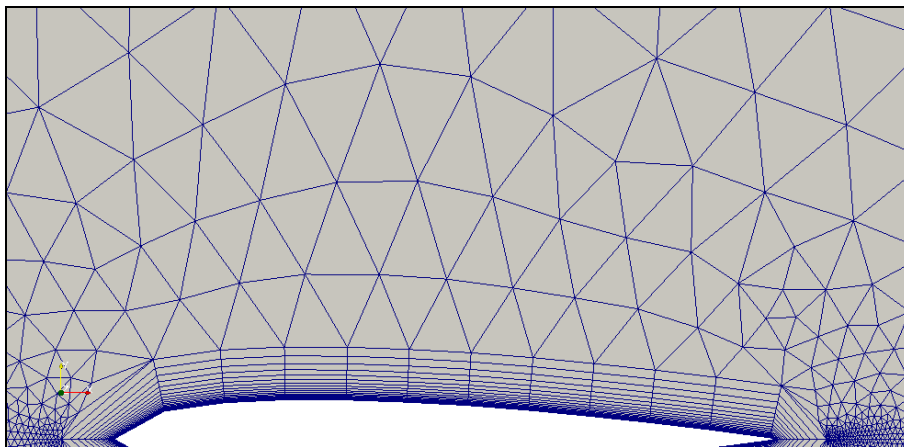


Figura 28: Detalle de mallado planteado en OpenFOAM

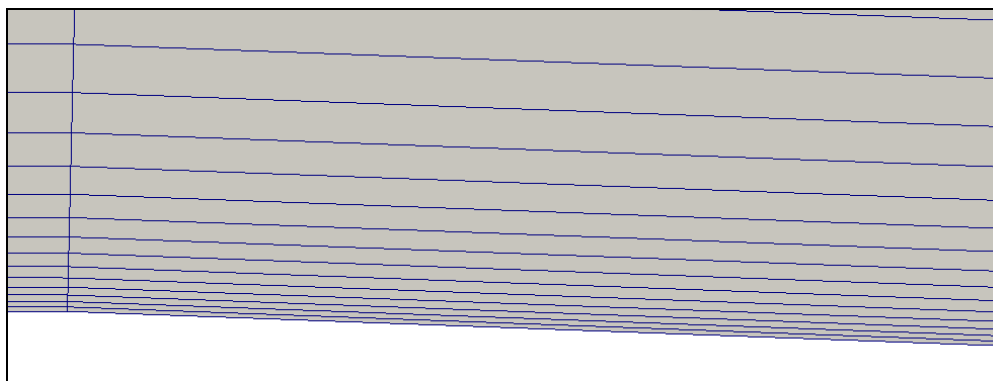


Figura 29: Detalle 2 de mallado planteado en OpenFOAM

Es necesario extruir el mallado para que sea un volumen. Antes de ello, se importan los grupos hechos en la geometría (botón derecho a Mesh > *Create Groups from Geometry*) La extrusión se realiza en *Modification > Extrusion*. Se selecciona una profundidad de 0,1 m. El resultado es que a parte del mallado también se ha modificado los grupos importados. Únicamente son necesarias las caras que engloban el *inlet*, el perfil, el *outlet*, y las paredes *front* y *back*, correspondientes al frente y la parte de detrás del volumen. Se nombran de esta manera y el resto se deben borrar.

El mallado resultante tiene 9226 elementos (es un mallado bastante sencillo), 480 de los cuales son hexaedros y el resto son prismas. Para exportar el mallado a OpenFOAM se debe seleccionar *File > Export > UNV File*. Ya en el terminal, en el directorio del problema se realiza el comando *ideasUnvToFoam [nombre del archivo.unv]* para guardar toda la información de la malla en la carpeta *polyMesh*.

Para terminar, se realiza un estudio de calidad del mallado (en el terminal ejecutando *checkMesh*), donde se observa (Figura 30 y Figura 31) que existen diversos elementos con una relación de aspecto elevada. Estos son susceptibles de ser los correspondientes a los elementos hexaédricos, pero debido a que están orientados en consonancia a como circula el fluido no debería ser una causa de fallo a la hora de encontrar una solución.

```
Time = 0

Mesh stats
  points:      10308
  internal points: 0
  faces:      32832
  internal faces: 13778
  cells:      9226
  faces per cell: 5.05203
  boundary patches: 5
  point zones: 0
  face zones: 0
  cell zones: 0

Overall number of cells of each type:
  hexahedra: 480
  prisms: 8746
  wedges: 0
  pyramids: 0
  tet wedges: 0
  tetrahedra: 0
  polyhedra: 0

Checking topology...
  Boundary definition OK.
  Cell to face addressing OK.
  Point usage OK.
  Upper triangular ordering OK.
  Face vertices OK.
  Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
  Patch      Faces    Points  Surface topology
  back       9226    5154   ok (non-closed singly connected)
  inlet      378     758    ok (non-closed singly connected)
  outlet     200     402    ok (non-closed singly connected)
  airfoil    24      48     ok (non-closed singly connected)
  front      9226    5154   ok (non-closed singly connected)

Checking geometry...
  Overall domain bounding box (-5 -10 0) (10 10 0.1)
  Mesh has 2 geometric (non-empty/wedge) directions (1 1 0)
  Mesh has 2 solution (non-empty) directions (1 1 0)
  All edges aligned with or perpendicular to non-empty directions.
  Boundary openness (7.82334e-18 2.99233e-20 2.11447e-16) OK.
```

Figura 30: Resultados de la ejecución del comando checkMesh (1/2)

```

Checking patch topology for multiply connected surfaces...
Patch      Faces    Points  Surface topology
back       9226    5154   ok (non-closed singly connected)
inlet      378     758    ok (non-closed singly connected)
outlet     200     402    ok (non-closed singly connected)
airfoil    24      48     ok (non-closed singly connected)
front      9226    5154   ok (non-closed singly connected)

Checking geometry...
Overall domain bounding box (-5 -10 0) (10 10 0.1)
Mesh has 2 geometric (non-empty/wedge) directions (1 1 0)
Mesh has 2 solution (non-empty) directions (1 1 0)
All edges aligned with or perpendicular to non-empty directions.
Boundary openness (7.82334e-18 2.99233e-20 2.11447e-16) OK.
Max cell openness = 5.06655e-15 OK.
Max aspect ratio = 117.022 OK.
Minimum face area = 1.88019e-06. Maximum face area = 0.184857. Face area magnitudes OK.
Min volume = 1.88019e-07. Max volume = 0.0184857. Total volume = 19.9917.
Cell volumes OK.
Mesh non-orthogonality Max: 71.9119 average: 10.1726
*Number of severely non-orthogonal (> 70 degrees) faces: 4.
Non-orthogonality check OK.
<<Writing 4 non-orthogonal faces to set nonOrthoFaces
Face pyramids OK.
***Max skewness = 7.03586, 15 highly skew faces detected which may impair the quality of the results
<<Writing 15 skew faces to set skewFaces
Coupled point location match (average 0) OK.

Failed 1 mesh checks.

End

```

Figura 31: Resultados de la ejecución del comando `checkMesh` (2/2)

El mallado para ANSYS es directamente importado del archivo de OpenFOAM.

7.2.3. Set-up y solución

Las condiciones de contorno son las siguientes:

- En el *inlet* la velocidad del aire es de 10m/s. Para indicar la inclinación de 2° respecto la coordenada X, se descompone en los ejes X,Y, obteniendo un valor de $(V_x, V_y) = (9,99939, 0,34899)$
- La presión del *outlet* es atmosférica.
- A las paredes frontal y trasera se les aplica la condición *empty* (como en el primer estudio).
- El perfil de ala es tratada como una pared estática sin deslizamiento (condición *noSlip*).

Se determinan las propiedades de la pared describiéndolas de la manera que indica la Figura 32 en el fichero *boundary* dentro de la carpeta *polyMesh*, y la velocidad y la presión de la manera que indican la Figura 33 y Figura 34 dentro de los respectivos ficheros de *u* y *p* en la carpeta 0. El resto de ficheros de 0 se dejan por defecto, siempre teniendo en cuenta que las paredes tengan el mismo nombre y las del frente del volumen y la parte de detrás, así como el perfil, ya que en el ejemplo que se ha cogido de base tienen nombres diferentes.

```
17
18 5
19 (
20   back
21   {
22     type          empty;
23     inGroups      1(empty);
24     nFaces        9226;
25     startFace     13778;
26   }
27   inlet
28   {
29     type          patch;
30     nFaces        378;
31     startFace     23004;
32   }
33   outlet
34   {
35     type          patch;
36     nFaces        200;
37     startFace     23382;
38   }
39   perfil
40   {
41     type          wall;
42     nFaces        24;
43     startFace     23582;
44   }
45   front
46   {
47     type          empty;
48     inGroups      1(empty);
49     nFaces        9226;
50     startFace     23606;
51   }
52 )
53
54 // ***** //
55
```

Figura 32: Parámetros del fichero *boundary*

```

15 // * * * * * //
16
17 dimensions      [0 1 -1 0 0 0];
18
19 internalField   uniform (0 0 0);
20
21 boundaryField
22 {
23     inlet
24     {
25         type      fixedValue;
26         value     uniform (9.99390827 0.348994967 0);
27     }
28
29     outlet
30     {
31         type      zeroGradient;
32     }
33
34     perfil
35     {
36         type      noSlip;
37     }
38
39     front
40     {
41         type      empty;
42     }
43     back
44     {
45         type      empty;
46     }
47 }
48
49 // * * * * * //
50

```

Figura 33: Fichero U de la carpeta 0

```

15 // * * * * * //
16
17 dimensions      [0 2 -2 0 0 0];
18
19 internalField   uniform 0;
20
21 boundaryField
22 {
23     inlet
24     {
25         type      zeroGradient;
26     }
27
28     outlet
29     {
30         type      fixedValue;
31         value     uniform 0;
32     }
33
34     perfil
35     {
36         type      zeroGradient;
37     }
38
39     front
40     {
41         type      empty;
42     }
43     back
44     {
45         type      empty;
46     }
47 }
48
49 // * * * * * //

```

Figura 34: Fichero P de la carpeta 0

Las propiedades del fluido son las mismas que el primer caso, aunque el número de Reynolds es diferente, debido a que la velocidad y la longitud característica (cuerda del perfil, 0.1m) han cambiado. El número de Reynolds resultante es $1 \cdot 10^7$, por lo que el flujo es claramente turbulento. Para indicar estas propiedades el procedimiento es el mismo que en el primer estudio.

También se hace un estudio estacionario, incompresible (ya que el número de Mach es muy inferior a 0,3), no se tiene en cuenta el efecto de la temperatura, y la presión de referencia es la atmosférica.

Por los mismos motivos que el estudio anterior, el modelo de turbulencias es k- ϵ y se aplican funciones de pared estándar. También por la misma causa, el algoritmo de solución está basado en la presión, pero se escoge en OpenFOAM concretamente el PIMPLE. Es una mezcla entre el algoritmo PISO y el SIMPLE (*Semi-Implicit Method for Pressure Linked Equations*). El SIMPLE es un algoritmo enfocado a casos estacionarios. El motivo de elección es que el intervalo de integración en este algoritmo es variable, dependiendo de la necesidad, por lo que puede hacer que sea mucho más grande en casos en los cuales con el PISO estaría fijado, por lo que ayuda a reducir el tiempo computacional. En ANSYS el algoritmo usado sigue siendo el PISO. Estas condiciones ya vienen por defecto en el tutorial de base escogido, por lo que no es necesario modificar ningún parámetro en este aspecto.

Antes de poder acceder a la solución queda modificar los parámetros de la solución *controlDict* en el fichero *system* (Figura 35). Como el incremento de tiempo es variable, no es necesario fijar un número de Courant máximo, como en el primer caso, para que la solución no diverja, pero sí que se escoge uno orientativo para iniciar el problema. Para ello se usa la Ecuación 18 donde el incremento de espacio escogido se aproxima como la raíz cúbica del volumen del elemento más pequeño en la malla (Figura 31) y la velocidad se elige de 10m/s, como en el *inlet*. El resultado es de un incremento de tiempo de 0,0005 s. El tiempo final de simulación es de 40s, lo que dan un total de 40000 iteraciones, aunque solo se registran 40 (se guardan los datos cada 0,5 s).


```
16 // ***** //
17
18 application    pimpleFoam;
19
20 startFrom      latestTime;
21
22 startTime      0;
23
24 stopAt         endTime;
25
26 endTime        20;
27
28 deltaT         0.0005;
29
30 writeControl   adjustableRunTime;
31
32 writeInterval  0.5;
33
34 purgeWrite     0;
35
36 writeFormat    ascii;
37
38 writePrecision 6;
39
40 writeCompression off;
41
42 timeFormat     general;
43
44 timePrecision  6;
45
46 runTimeModifiable yes;
47
48 adjustTimeStep yes;
49
50 maxCo          5;
51
52 // ***** //
```

Figura 35: Parámetros de controlDict

Se procede a la solución ejecutando en el terminal el comando *pimpleFoam*.

La convergencia es monitorizada para los valores de los coeficientes de estudio. Los resultados se obtienen mediante las herramientas de post-procesado en Paraview.

7.2.4. Post-procesado, resultado y conclusiones

Para obtener estos datos de coeficientes lo primero necesario es mostrar únicamente la superficie del perfil del ala. Para ello, en las propiedades del caso base del cuadro de partes de la malla únicamente se debe mostrar el perfil. Después se generan los vectores normales (*Filter > Generate Surface Normals*) a cada una de las superficies que generan el perfil mallado (esto sirve para poder indicar la dirección para las fuerzas). Una vez generadas las normales, mediante dos calculadoras se realizan las siguientes operaciones (Ecuación 20 y Ecuación 21):

$$C_D = \frac{2F_D}{\rho V^2 A} = \frac{p \cdot normalsX \cdot 2}{100 \cdot 0.1}$$

Ecuación 20: Cálculo del coeficiente de arrastre aplicado en la calculadora de Paraview

$$C_L = \frac{2F_L}{\rho V^2 A} = \frac{p \cdot normalsY \cdot 2}{100 \cdot 0.1}$$

Ecuación 21: Cálculo del coeficiente de sustentación aplicado en la calculadora de Paraview

No se aplica la densidad al cálculo ya que está incluida dentro de lo que el programa considera como presión. Una vez extraídos los resultados en cada tiempo se grafican y el resultado es el de la Figura 36:

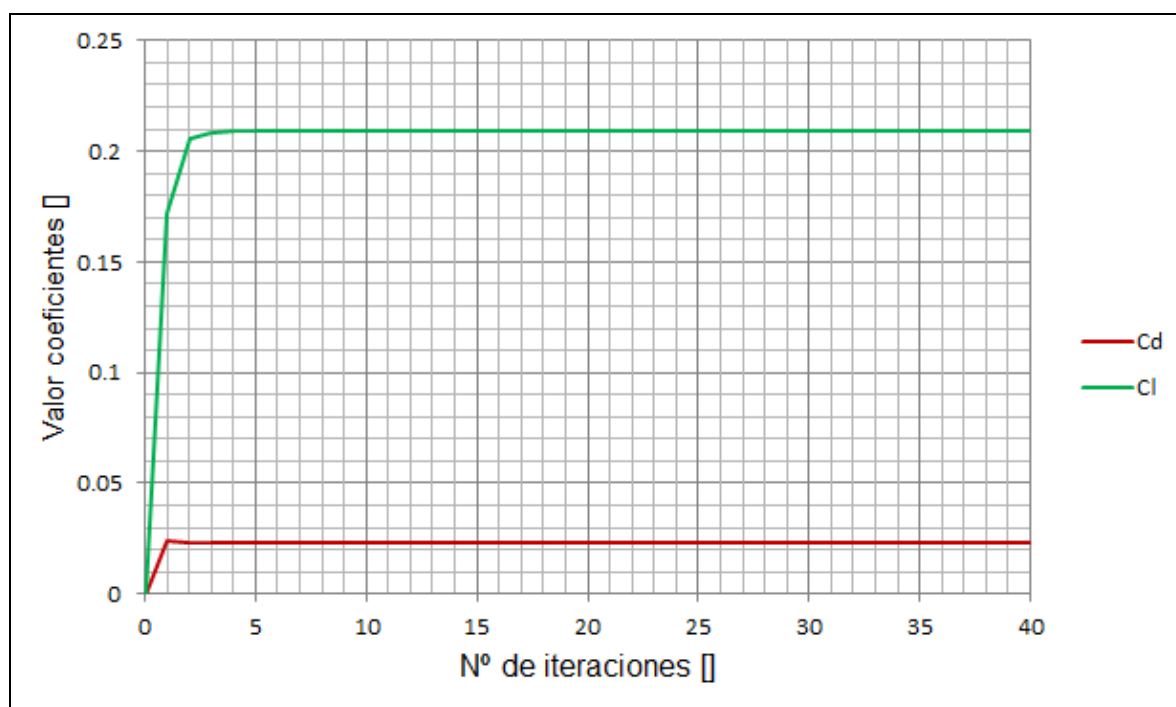


Figura 36: Monitorización de los coeficientes C_D y C_L en OpenFOAM

Como se puede comprobar, no era necesario un número tan alto de iteraciones, puesto que ha convergido antes de llegar a 5 iteraciones registradas en carpetas (es decir, antes de 10000 totales).

Comparado con los resultados obtenidos por ANSYS (Figura 37):

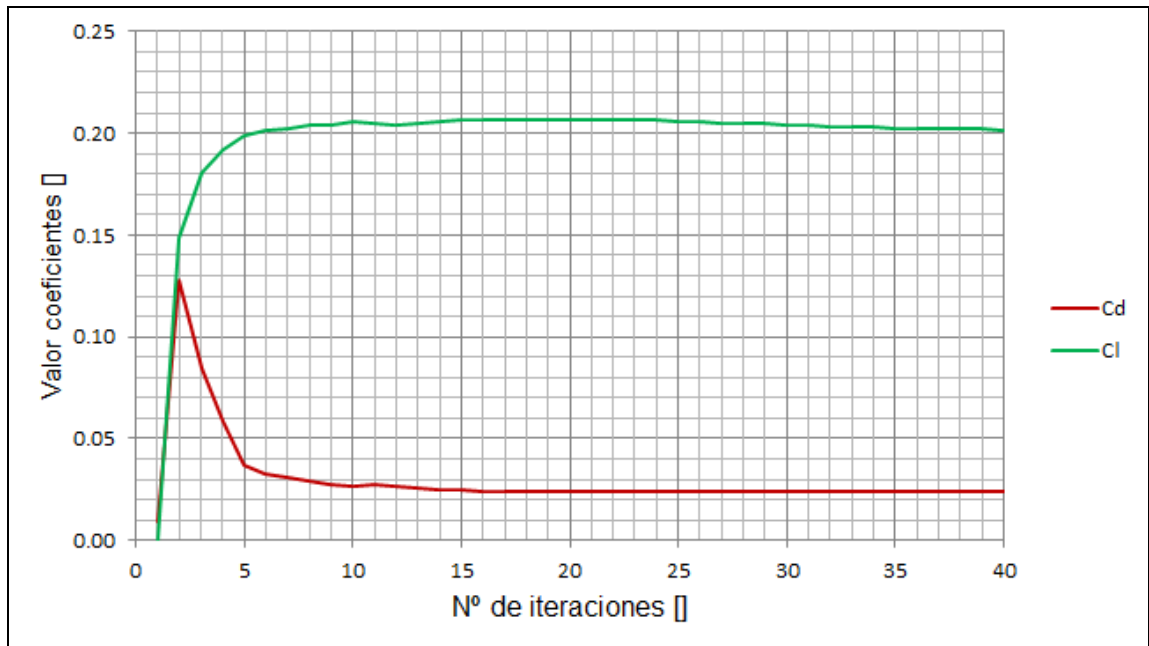


Figura 37: Monitorización de los coeficientes C_D y C_L en ANSYS

Se puede observar que los resultados son muy parecidos, pero que en ANSYS no se han tenido que hacer tantas iteraciones para llegar al resultado. Esto puede ser debido a las condiciones de los algoritmos impuestas en cada uno de ellos, ya que el resto de parámetros se mantienen de la misma forma. Ambos son resultados coherentes, ya que al comparar los datos con algunos datos experimentales [10] resultan situados dentro del rango. Se puede entender que este ala está inclinada con un ángulo el cual le permite ganar una cantidad de sustentación significativa sin excesivo arrastre.

A continuación en la Figura 38, Figura 39, Figura 40 y Figura 41 se presentan los resultados de las líneas de corriente (*Filters > streamlines* en OpenFOAM) y los mapas de presión en OpenFOAM y ANSYS:

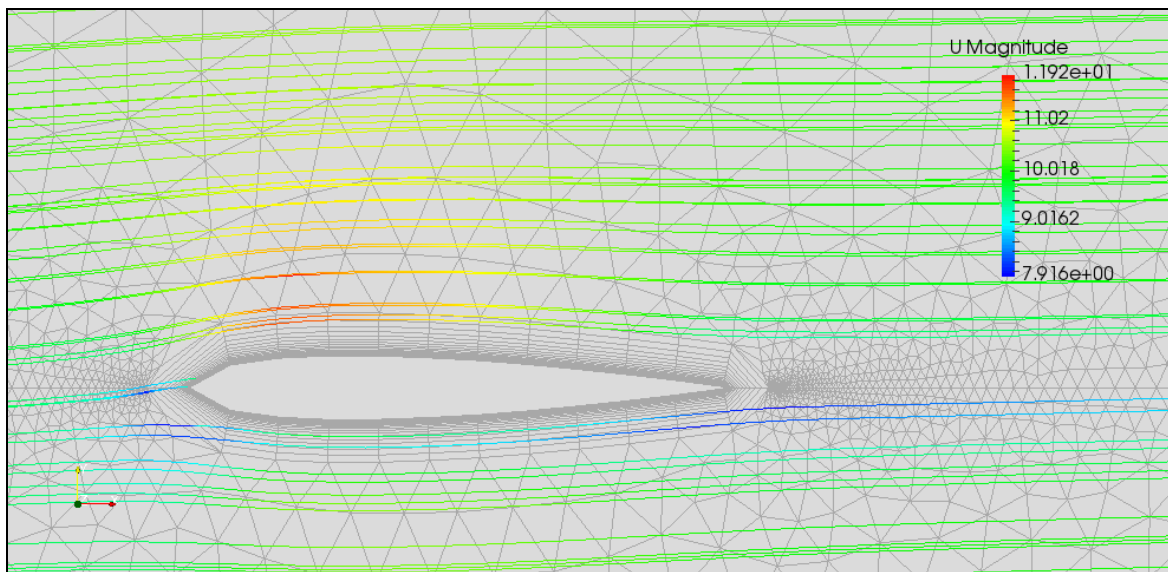


Figura 38: Streamline en OpenFOAM

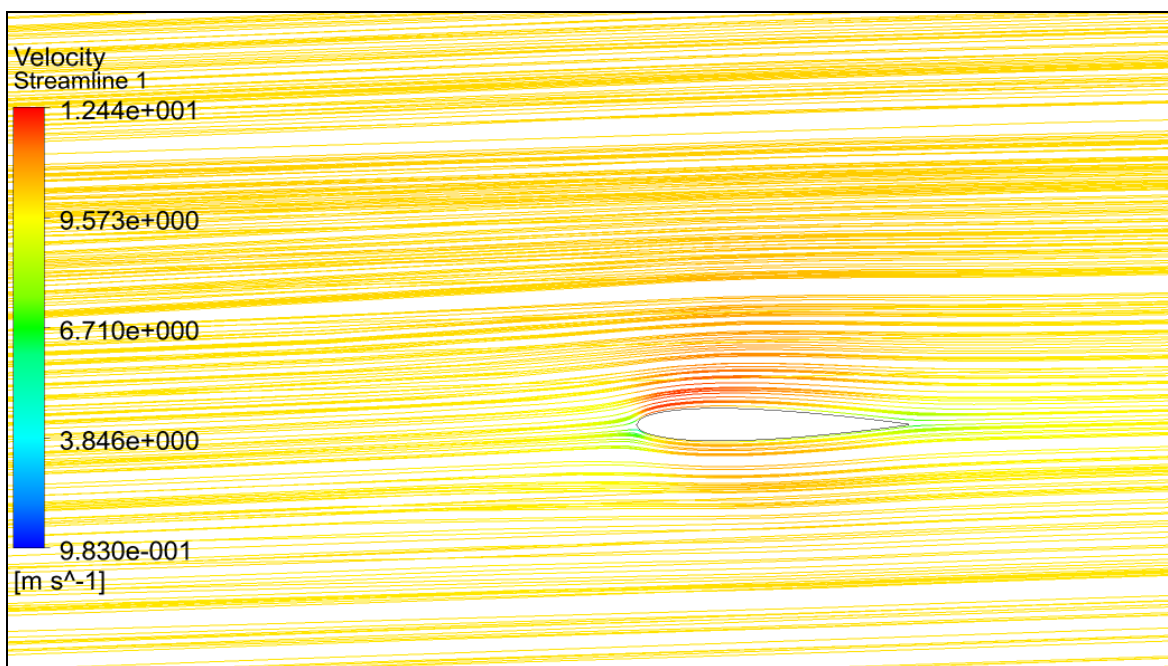


Figura 39: Streamline en ANSYS

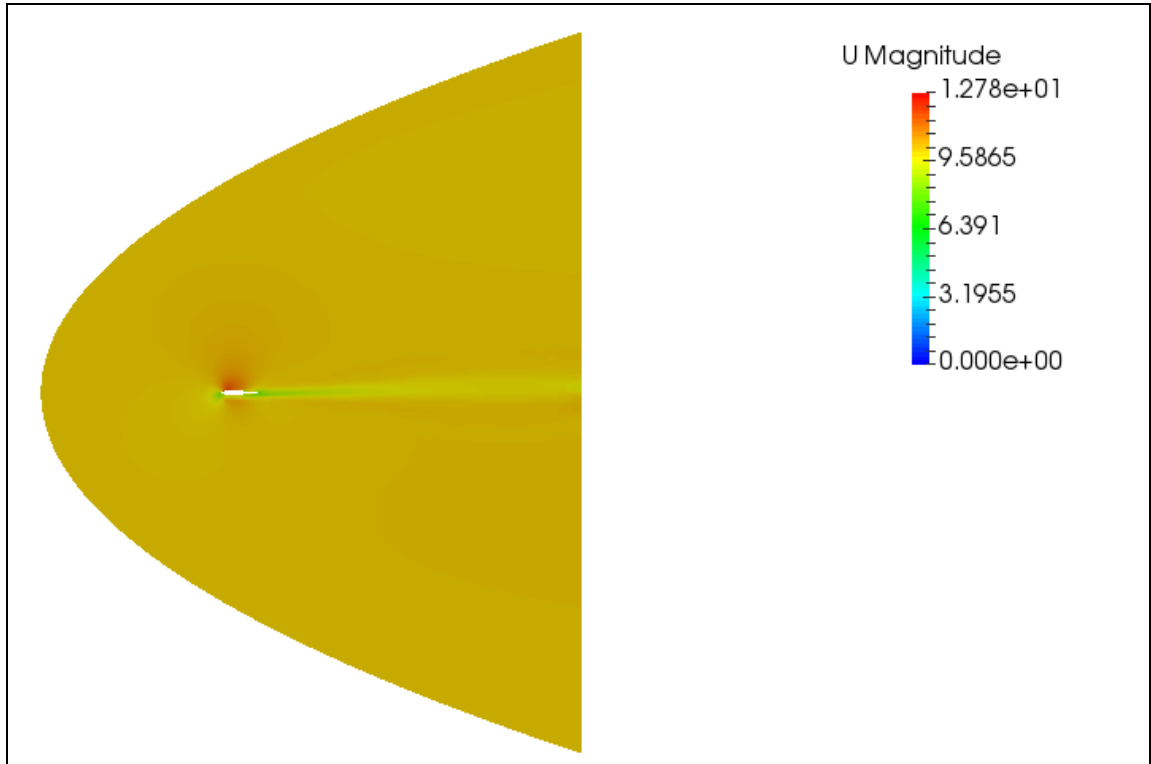


Figura 40: Mapa de velocidades en OpenFOAM

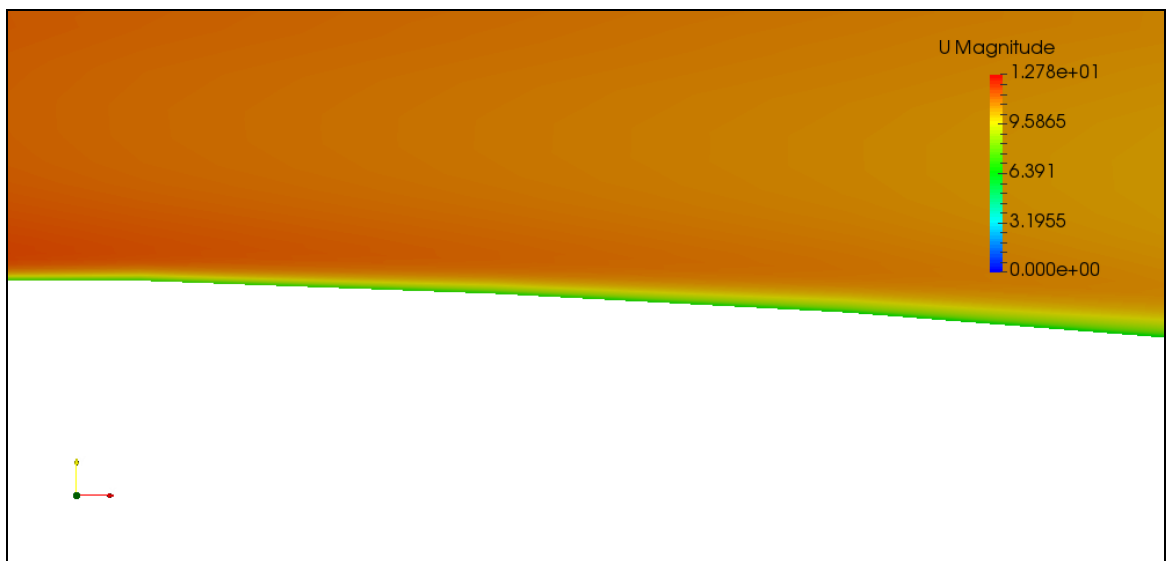


Figura 41: Detalle 2 de mapa de velocidades en OpenFOAM

Se pueden ver claramente en las líneas de corriente la diferencia de velocidad en la parte superior e inferior del perfil, debido a la inclinación dada que hace que el punto de estancamiento (donde la velocidad es cero) se sitúe por debajo del plano de simetría del perfil. En el caso de los mapas de velocidad también se puede ver, así como las velocidades más bajas que se producen detrás del ala, relacionadas con la generación de

turbulencias en esa zona. Se puede ver en detalle en la Figura 41 la variación del perfil de velocidad correspondiente a las regiones más cercanas al ala, siendo 0 el valor de la velocidad justo en el punto de contacto del fluido con ella. En comparación con los valores de ANSYS se puede observar como hay diferencias con respecto los valores máximos y mínimos que pueden ser debidas al mismo motivo descrito anteriormente.

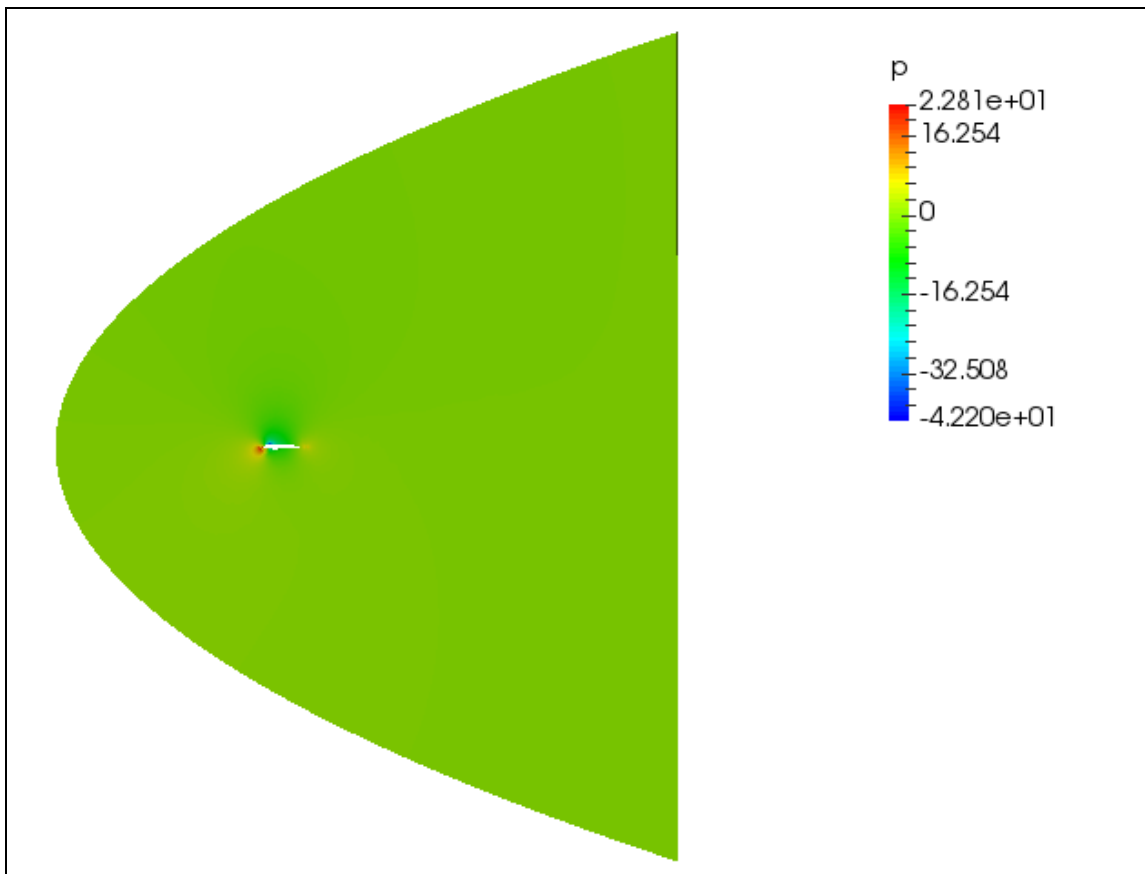


Figura 42: Mapa de presiones en OpenFOAM

Respecto el mapa de presiones (Figura 42), se puede visualizar como la presión en el punto de estancamiento es alta, y es más reducida en el punto de más velocidad. Si se ve en detalle la región próxima al perfil (Figura 43):

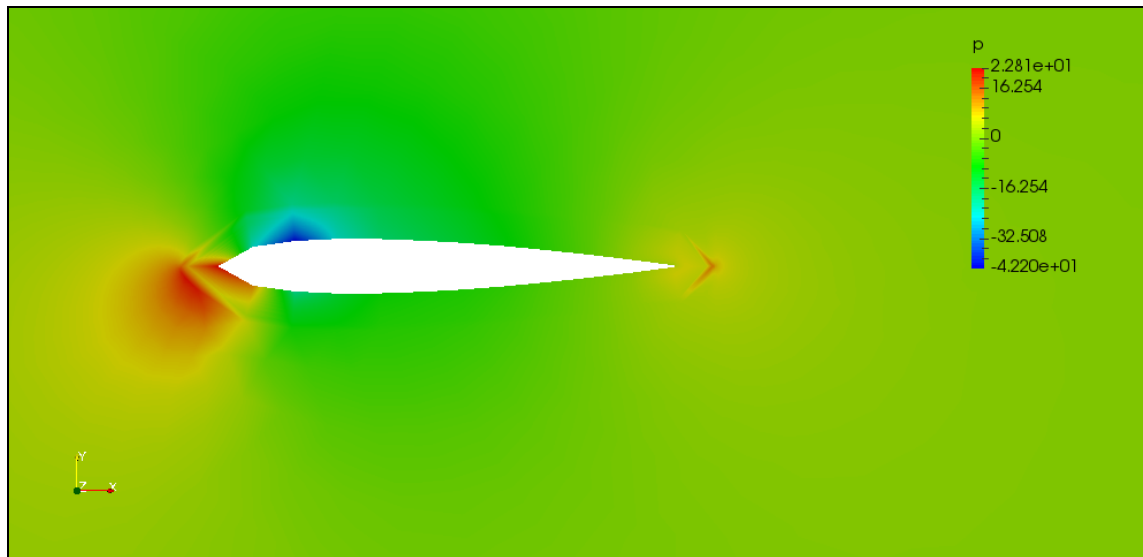


Figura 43: Detalle del mapa de presiones en OpenFOAM

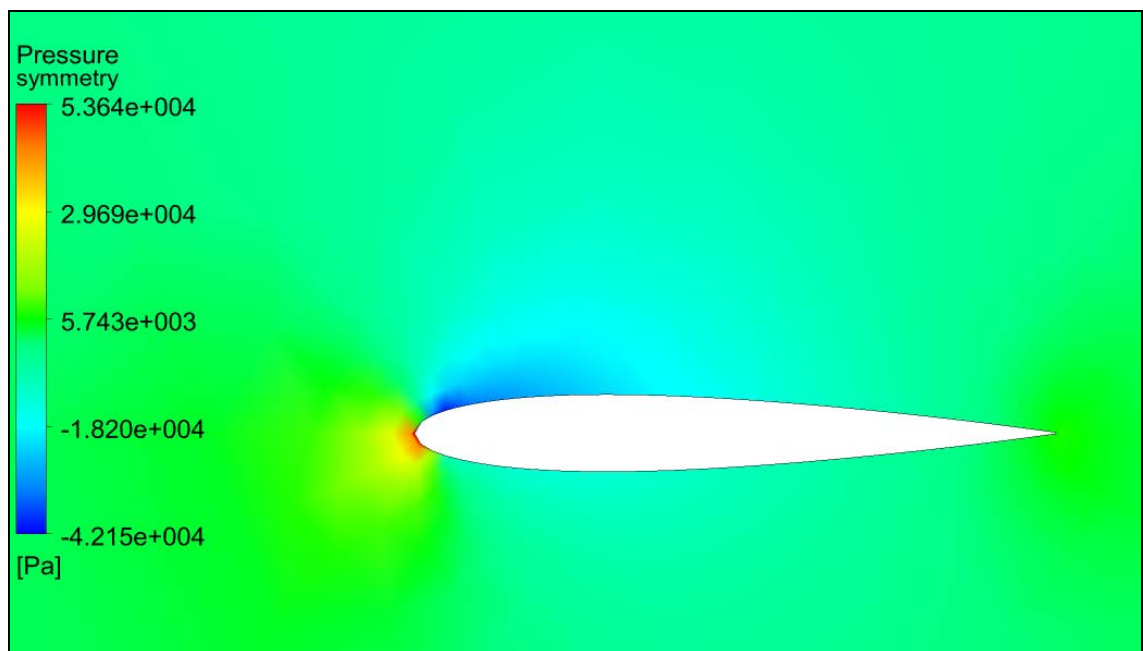


Figura 44: Detalle del mapa de presiones en ANSYS

Se puede ver tanto en la zona anterior como posterior del ala unos cambios de presión bruscos, que no se ven en ANSYS (Figura 44). Esto es debido a la transición brusca que existe entre los elementos hexaédricos de la región más próxima al perfil y los prismas correspondiente a la más externa. Este fenómeno puede tener como consecuencia una generación de error en los valores resultado.

8. Valoración del software libre para el uso en simulación numérica

Cada tipo de software tiene sus ventajas e inconvenientes. En el caso de los *softwares* libres, aunque hay algunas páginas con cierta información sobre tutoriales básicos para realizar ciertos casos, no hay suficientes manuales de referencia, guías de usuario o métodos paso a paso para que un usuario sin conocimiento alguno pueda comenzar a aprender. En este campo, los programas comerciales como ANSYS poseen una cantidad muy superior de información, aunque también se debe a la diferencia de popularidad entre uno y otro.

Respecto a los parámetros de mallado y geometría, OpenFOAM se puede utilizar para casos sencillos, y aunque hay opciones que sirven para geometrías más complejas, es más conveniente utilizar SALOME. Este motivo supone la adición de una herramienta adicional al proceso de simulación, y genera una inconveniencia. En el caso de ANSYS todas las herramientas están englobadas en un único espacio de trabajo (llamado Workbench), y simplifica el uso de geometrías o estructuras malladas para simulaciones distintas (Figura 45).

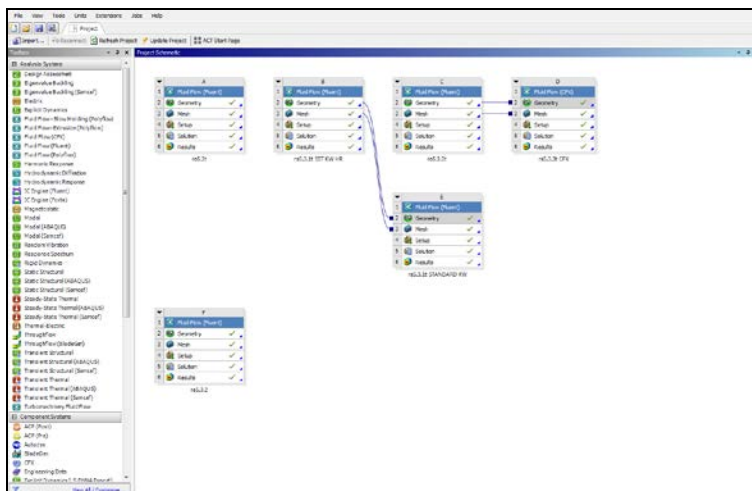


Figura 45: Interfaz Workbench en ANSYS

ANSYS y SALOME disponen de muchas opciones que permiten realizar estructuras diferentes. En el caso del mallado, en ambos se puede seleccionar diferentes elementos, tamaños y formas. También se puede mallar en regiones determinadas de una forma diferente y aplicar condiciones específicas a estas zonas. En ANSYS dispone de una opción de mallado predeterminado, que si bien no es la más óptima, puede servir para una opción rápida o aproximada. En el caso de que se quiera mejorar la calidad, se pueden

modificar los parámetros por defecto. En SALOME es una opción que sería útil, ya que en este programa se debe seleccionar todas las opciones antes de computar el mallado. Esto puede hacer que un usuario que no tiene pleno conocimiento de cualquier opción introducida puede acabar con un resultado final no deseado. Tampoco es intuitivo la forma en que en SALOME se debe generar y agrupar regiones con el objetivo de aplicar un algoritmo de mallado o una condición de contorno.

Referente a la preparación de la simulación, OpenFOAM contiene la información necesaria en carpetas, donde el usuario a mano debe añadirlas con su correspondiente información o modificarlas a partir de un tutorial (que se obtiene juntamente con la instalación del programa). En cada fichero editable la información se debe proporcionar se forma adecuada, haciendo que cualquier error cometido de escritura provoque que la solución no se pueda computar. A veces el mensaje obtenido de respuesta al intentar ejecutar la solución puede ayudar a localizar el error. Aun así, la mayoría de estos errores no son comunes en softwares más cerrados, porque la introducción de datos (y de fallos) es mucho más limitada.

El proceso de solución en ANSYS grafica por defecto los residuos y otros parámetros de interés en tiempo real, y es una opción que podría ser útil también en OpenFOAM. En el caso del último sí que muestra los valores en el terminal pero cuando la solución es rápida no se pueden observar con claridad. Una vez hecha la solución, OpenFOAM crea carpetas con los valores de todas las variables para todas las iteraciones. Esto puede ser útil para la consulta de un valor en concreto, aunque se puede hacer con facilidad en el post-procesado en Paraview.

La representación gráfica de resultados en ANSYS y Paraview es semejante, aunque como en el caso de la geometría y el mallado sería más oportuno recoger todas las herramientas en un único programa.

En general, OpenFOAM no se enfoca a ser una herramienta para hacer simulaciones en la que esté todo integrado en una única interfaz. En cambio, es un software de simulación con más libertad para modificar cualquier parámetro en cualquier etapa de la simulación. Esto es un inconveniente para un usuario que busca realizar una simulación sencilla, pero es una ventaja para un estudio profundo en el que todos los parámetros se quieran controlar.

9. Impacto ambiental

La elaboración de este proyecto ha sido completamente teórica. La perturbación al medio ambiente provocado por este proyecto ha sido nula, a excepción del consumo energético que tiene hacer uso de los ordenadores durante la duración el estudio. La electricidad consumida por estos ordenadores, debido a su procedencia, mayoritariamente proviene de una explotación no sostenible, es un factor a tener en cuenta.

Estos ordenadores (un portátil y uno de sobremesa, y varios de la universidad) no han sido dedicados únicamente a este trabajo, se han utilizado para otros motivos. Por este motivo la creación de estos ordenadores no es considerado que forme parte del impacto ambiental del trabajo, únicamente su uso durante el proyecto.

Como consideración, la facilidad de uso de herramientas de simulación numérica sirve como una opción viable para evitar la realización de estudios experimentales, en los que el uso de materiales y energético probablemente habría sido más grande. Es decir, se puede considerar el desarrollo de este proyecto una ayuda para evitar un impacto ambiental mayor.

Un indicador es el número de CO₂ emitido para generar la electricidad suficiente para el proyecto (Tabla 2). Para ello, se ha estimado la electricidad consumida de ambos ordenadores y la luz del entorno de trabajo, y se ha obtenido el factor de emisión a partir de la información disponible en un calculador *online* [12].

	Consumo (kW)	Factor de emisión (kg/kWh)	CO ₂ equivalente (kg)
Electricidad	0,7	0,385	0,2695

Tabla 2: Cálculo de CO₂ generado durante el proyecto

10. Planificación

La duración final del trabajo se ha determinado por la disponibilidad en ambos cuatrimestres. Los intervalos de tiempo son muy grandes por la imposibilidad de dedicación completa. El proyecto se puede dividir en las siguientes fases:

- Fase introductoria: primeramente se ha debido determinar los objetivos y el alcance de este trabajo.
- Fase de documentación teórica: ha sido necesario aprender los contenidos teóricos de mecánica de fluidos necesarios para el estudio de ambos problemas.
- Fase de aprendizaje de *software*: una vez los conceptos teóricos asimilados, ha sido necesario entender y saber utilizar todo el *software* utilizado en este trabajo.
- Desarrollo del primer estudio en ANSYS: esta fase se puede desglosar también en la preparación de la geometría y mallado, set-up, solución, obtención de resultados mediante el post-procesado y elaboración de conclusiones.
- Desarrollo del primer estudio en OpenFOAM: la división de esta fase es idéntica a la anterior
- Desarrollo del segundo estudio en ANSYS
- Desarrollo del segundo estudio en OpenFOAM
- Redacción de la memoria

Años	2018				2019				
	Meses	9	10	11	12	1	2	3	4
<i>Fase introductoria</i>	█								
<i>Fase de documentación teórica</i>	█	█							
<i>Fase de aprendizaje de software</i>		█	█	█					
<i>Desarrollo del primer estudio en ANSYS</i>			█	█	█				
<i>Desarrollo del primer estudio en OpenFOAM</i>			█	█	█				
<i>Desarrollo del segundo estudio en ANSYS</i>					█	█	█		
<i>Desarrollo del segundo estudio en OpenFOAM</i>						█	█	█	█
<i>Redacción de la memoria</i>				█	█	█	█	█	█

Tabla 3: Planificación

11. Presupuesto

El presupuesto del proyecto, al no tratarse de ningún tipo de fabricación de producto o prototipo, engloba únicamente costes de ingeniería, energéticos y licencias de programas.

Se presupone un coste de ingeniería que se facturaría en caso de vender el proyecto, y que las licencias y consumos anuales son amortizados en otros proyectos de manera constante durante el año, ya que lo más probable es que se utilicen estos recursos para más de un caso.

Concepto	Precio (€/kw·h)	Electricidad consumida (kW)	Precio (€/año)	Precio (€/h)	Duración (h)	Precio total (€)
Material de oficina	-	-	-	-	-	20,00
Licencia de ANSYS	-		5000,00	0,571	300	2853,88
Licencia de Office 365	-		70,00	0,008	300	2,40
Consumo eléctrico	0,12	0,45	-	-	300	16,20
Consumo de internet	-		240,00	0,027	300	6,55
Costes de ingeniería	-		-	50	300	15000
SUBTOTAL						17899,03
IVA (21%)						3758,80
TOTAL						21657,83

Tabla 4: Presupuesto del proyecto

El cálculo de la licencia de ANSYS es estimado, ya que su precio depende sustancialmente de si el destinatario es un cliente, empresa y universidad. También depende de la versión adquirida.

Conclusiones

Respecto las dos simulaciones realizadas, cabe decir que los resultados obtenidos por OpenFOAM han sido semejantes a los de ANSYS. Gracias a ello se puede considerar como bueno el uso dado al programa, y por lo tanto, la capacidad que tiene el programa de ofrecer resultados coherentes a usuarios con un conocimiento básico de esta herramienta.

Las simulaciones de este trabajo han sido partiendo de geometrías, propiedades e hipótesis relativamente sencillas. Esto ha sido causa de la capacidad computacional limitada que disponía el ordenador con el que se realizaban las simulaciones. El mallado, por ejemplo, precisa de una mayor calidad y estudio para poder obtener una calidad óptima de resultados sin recurrir a una gran cantidad de recursos computacionales. Aun así, este trabajo ayuda a comprender el funcionamiento de estos programas usados, y por este motivo puede servir de base para futuros proyectos donde se requiera un estudio de casos mucho más completos. Con ello, puede encaminarse con más facilidad la introducción de nuevos módulos. Un ejemplo sería un caso de estudio de cavitación.

Para finalizar, uno de los propósitos de este trabajo es dar una alternativa a nivel académico (como en el caso de las prácticas de Mecánica de Fluidos) a los estudios de CFD con ANSYS, para poder tener libertad de uso y mucho más fácil acceso. Como conclusión, se ha demostrado esta que esta opción es viable, y que la introducción de esta herramienta en la universidad pueda tener como consecuencia un aumento de su uso en la industria.

Agradecimientos

Primeramente me gustaría agradecer a mi profesor y tutor de este trabajo, Xavier Escaler Puigoriol, por su supervisión y ayuda durante todo el transcurso del proyecto. Gracias a él he podido conocer la existencia de alternativas a *software* comercial, y he podido ver la aplicación necesaria de las simulaciones computacionales de fluidos a la industria.

Además doy las gracias a mis compañeros de equipo de ETSEIB Racing, con los que he podido aprender y aplicar conocimientos sobre CFD en elementos de una motocicleta de competición y me ha motivado a seguir aprendiendo conocimientos en este campo. En especial, agradezco a Carlos Picó transferirme sus conocimientos sobre la metodología de realización de simulaciones.

Agradezco, por último, a mi familia y a mis amigos, por su paciencia y apoyo continuo con este trabajo y con el resto del grado.

Bibliografía

Referencias bibliográficas

- [1] *What is free software?- GNU Project - Free Software Foundation*. Visitado en 2018. Disponible en:
<<https://www.gnu.org/philosophy/free-sw.en.html>>
- [2] *Engineering Simulation & 3D Design Software | ANSYS*. Visitado en 2018. Disponible en:
<<https://www.ansys.com/>>
- [3] UPC. *Libro de apuntes de la asignatura de Mecánica de Fluidos (2016)*. Barcelona.
- [4] *SALOME Platform*. Visitado en 2018. Disponible en:
<<https://www.salome-platform.org/>>
- [5] *Law of the wall*. Visitado en 2018. Disponible en:
<<http://brennen.caltech.edu/fluidbook/basicfluidynamics/turbulence/lawofthewall.pdf>>
- [6] *P04R Streamline Test*. Visitado en 2019. Disponible en:
<<https://www.youtube.com/watch?v=0aHqgLGPUe0>>
- [7] The community. *Xubuntu*. Visitado en 2018. Disponible en:
<<https://xubuntu.org/>>
- [8] *Code_Aster*. Visitado en 2018. Disponible en:
<<https://www.code-aster.org/spip.php?rubrique2>>
- [9] *OpenFOAM - Official home of The Open Source Computational Fluid Dynamics (CFD) Toolbox*. Visitado en 2018. Disponible en:
<<https://www.openfoam.com/>>
- [10] *NACA 0012 AIRFOILS (n0012-il)*. Visitado en 2018. Disponible en:

<<http://airfoiltools.com/airfoil/details?airfoil=n0012-il>>

- [11] Dassault Systèmes. *Dassault Systèmes: 3D Content Central*. Visitado en 2019. Disponible en:

<<https://www.3dcontentcentral.es/>>

- [12] Cámara de Zaragoza. *Calculador de emisiones*. Visitado en 2019. Disponible en:

<<https://www.camarazaragoza.com/>>

Bibliografía complementaria

- [1] White, F. (2010). *Mecánica de fluidos*. MacGraw-Hill.

- [2] *CFD Direct | The Architects of OpenFOAM*. Visitado en 2018. Disponible en:
<<https://cfd.direct/>>

- [3] *CFD Online Discussion Forums*. Visitado en 2018. Disponible en:

<<https://www.cfd-online.com/Forums/>>

- [4] *Basics Of Y Plus Boundary Layer And Wall Function In Turbulent Flows*. Visitado en 2018. Disponible en:

<https://www.learncax.com/knowledge-base/blog/by-category/cfd/basics-of-y-plus-boundary-layer-and-wall-function-in-turbulent-flows>

- [5] *Law of the wall -- CFD-Wiki, the free CFD reference*. Visitado en 2018. Disponible en:
<https://www.cfd-online.com/Wiki/Law_of_the_wall>

- [6] Escuela Superior de Ingenieros – Universidad de Navarra *Flujo Laminar y turbulento*. Visitado en 2018. Disponible en:

<<http://www4.tecnun.es/asignaturas/Fluidos1/WEBMF/Mecanica%20de%20Fluidos%20I/FAQMFI/FAQ12.html>>

- [7] *Ansys Documentation: Index of /Software/Ansys*. Visitado en 2018. Disponible en:

<<https://www.sharcnet.ca/Software/Ansys/>>

- [8] *OpenFOAM: User Guide: OpenFOAM®: Open source CFD*. Visitado en 2018. Disponible en:

<<https://www.openfoam.com/documentation/guides/latest/doc/index.html>>

[9] *OpenFOAM Folder Structure*. Visitado en 2018. Disponible en:

<http://www.cfdyna.com/Home/OpenFOAM/OpenFOAM_File_Struct.pdf>

[10] *Reynolds Average Navier-Stokes Equations*. Visitado en 2018. Disponible en:

<http://daad.wb.tu-harburg.de/fileadmin/BackUsersResources/Flood_Probability/2D/Stepfi-2D/pdf/Reynolds_average_Navier-Stokes_equation.pdf>

[11] NASA. *The Drag Coefficient*. Visitado en 2019. Disponible en:

<<https://www.grc.nasa.gov/www/k-12/airplane/dragco.html>>